

# DARwin

Dissimilarity Analysis and Representation  
for windows



with support from  
Generation Challenge Programme

Version 6

DARwin software

<http://darwin.cirad.fr>

written with Microsoft™ Visual Basic Studio.Net 2010  
Xavier PERRIER & J-P JACQUEMOUD-COLLET  
CIRAD - BIOS Department  
UMR AGAP - Genetic Improvement and Adaptation  
of Mediterranean and Tropical Plants  
Biomathematics team

Avenue Agropolis - TA A-108 / 03  
34398 - Montpellier cedex 5  
France

E-mail: [darwin@cirad.fr](mailto:darwin@cirad.fr)

Last updated 2014/10/20

[Send E-Mail to DARwin Team](#)

---

## Contents

<b>Introduction</b>	<b>5</b>
• Warning	5
• Installing DARwin6	5
<b>Overview</b>	<b>7</b>
<b>General features</b>	<b>7</b>
• DARwin file edition	7
• Graphical export	8
<b>Data and file formats</b>	<b>9</b>
• Data file .VAR components	10
▪ Single data	10
▪ Allelic data	11
▪ Sequence data	11
• Data file .AFT components	11
• Data file .DIS components	12
• Data file .ARB components	13
• Data file .DON components	14
<b>File menu</b>	<b>16</b>

• View File	16
• Importation / Exportation	16
▪ Import data matrix	16
▪ Import sequence	17
▪ Import dissimilarity	19
▪ Export dissimilarity	21
▪ Export dissimilarity as column	21
▪ Import tree	22
▪ Export tree	23
<b>Dissimilarity menu</b>	<b>25</b>
• Method	25
• Dissimilarity estimation	26
▪ Common options	27
▪ Dissimilarity for Single data	29
▪ Dissimilarity for Allelic data	32
▪ Dissimilarity for Sequence data	34
• Dissimilarity bargraph	40
• Dissimilarity extreme values	40
• Dissimilarity properties	41
• Metric index	41
• Euclidean index	42
• Furnas portraits	42
▪ Method	42
▪ Procedure	44
• Dissimilarity transformations	44
• Weighted sum of dissimilarities	46
<b>Factorial analysis</b>	<b>47</b>
• Method	47
• Analysis	48
• Graphical display	49
▪ Graph parameters	49
▪ Identification and illustration	49
▪ Graph exportation	50
<b>Tree construction</b>	<b>51</b>
• Method	51
▪ 'Neighbourhood' definition	51
▪ Dissimilarity updating	52
▪ Edge lengths	54
▪ Bootstraps	54
• Hierarchical tree	55
▪ Method	55
▪ Procedure	55
• Neighbor-Joining	56
▪ Method	56
▪ Procedure	57
• Scores	58
▪ Method	58
▪ Procedure	58
• Ordinal Neighbor-Joining	59

▪ Method	59
▪ Procedure	59
• Ordinal Scores	60
▪ Method	60
▪ Procedure	60
• Neighbor-Joining under topological constraints	61
▪ Method	61
▪ Applications	61
▪ Procedure	62
• Influential unit detection	63
▪ Method	63
▪ Procedure	63
<b>Trees...</b>	<b>65</b>
• Draw	65
▪ Tree draw toolbar	65
▪ Internal edge contraction	68
▪ Group definition	69
• Edge length bargraph	70
• Tree distance	71
• Fit criterion	71
• Least-squares re-estimation of edge lengths	72
▪ Method	72
▪ Procedure	72
• Pruning	73
• Grafting	73
▪ Method	73
▪ Procedure	74
• Max length sub tree	74
▪ Method	74
▪ Procedure	75
• Add a 2-degree node	78
• Remove all 2-degree nodes	78
• Reticulations	78
▪ Method	78
▪ Procedure	80
<b>Tree comparison</b>	<b>82</b>
• Consensus and tree distances	82
▪ Consensus methods	82
▪ 'Bipartition' distance between trees	83
▪ Procedure	84
• Maximum agreement sub-tree (MAST)	84
▪ Method	84
▪ MAST order as tree distance	85
▪ Maximal length MAST	86
▪ Procedure	86
• Quartet distance	87
▪ Method	87
▪ Procedure	88
<b>Disequilibrium</b>	<b>89</b>

• Background	89
• Method	89
▪ Max length sub tree	89
▪ Min SD subset	90
▪ Haplotypes	90
▪ Linkage versus structure disequilibrium	90
▪ Disequilibria on a set of loci	92
▪ Random samples as reference	92
▪ Allele richness	92
▪ Algorithm for Max length subtree	93
▪ Algorithm for Min SD subset	93
• Procedure	94
• Export to 'PHASE' software	99
▪ Method	99
▪ Procedure	99
• Import from 'PHASE' software	101
▪ Method	101
▪ Procedure	101

<b>Tools</b>	<b>102</b>
• Random 0/1 data	102
• Random dissimilarities	102
• Random tree	103
• Transpose data file	104
• Merge data files	105
• Single data correlations	107
• Re-label trees for common identifiers	108
• Pooling SSR alleles	110
▪ Method	110
▪ Marker selection	112
▪ Pooling alleles for a marker	114

<b>?</b>	<b>115</b>
• User's manual (PDF)	115
• How to cite DARwin	115

<b>Bibliography</b>	<b>115</b>
---------------------	------------

---

# Introduction

## ***Warning***

DARwin6 is provided free for use in research and education however the program is not open source.

IT IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE AUTHORS, THE CIRAD OR ITS DEPARTEMENT BIOS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

## ***Installing DARwin6***

DARwin6 can be directly downloaded from the web site <http://darwin.cirad.fr/>.

A registration is asked in order to maintain a list of DARwin users. This will enable us to assess the impact of this software and to defend renewed funding for new developments. In the same way, a list of applications using DARwin should be a convincing argument and we would appreciate that users send us references of articles, thesis, PHD...using DARwin.

You will be able to unregister at any time, and your email address will not be used for any purpose other than DARwin information.

### **System requirements**

DARwin software runs on Windows Operating System versions Windows 7 (32 and 64 bits), Windows Vista SP1 or later, Windows XP SP3, Windows XP SP2 x64 Edition, Windows Server 2008 (Server Core not supported), Windows Server 2008 R2 (Server Core supported with SP1 or later), Windows Server 2003 SP2.

DARwin uses Microsoft .Net Framework technology and is targeted on .Net Framework 4 Client profile. The setup procedure will automatically verify if a compatible version of the .Net Framework is installed on the computer. If the .Net Framework has to be installed or updated, the setup procedure will download automatically the best fitting version from Microsoft servers.

### **New features**

DARwin 6 is now developed on Visual Studio 2010 (Visual Basic .Net 2010). The software build runs on .Net Framework 4. This solution allows to exploit multi core exploitation on 32 bits (X86) and 64 bits (X64) operating systems.

Most of DARwin procedures are now parallelized to exploit multi core processors. Algorithms have been optimized to improve both accuracy and speed.



*DARwin hasn't real Dataset size limit. The size of the dataset which can be treated depends only of computer's physical characteristics: available memory and disk space.*



*Computing time in DARwin for most procedure will depend of computer performances: core number, frequency and Ram size.*

---

# Overview

DARwin is mainly focused on diversity structure description based on distance methods. It is organized in four logical parts: (i) dissimilarity measure, (ii) tree construction from dissimilarities, (iii) tree representation and edition, (iv) tree comparison. A fifth part is more specific and is devoted to sampling procedures to minimize disequilibria in a collection. Each part is independent and manages its own input data that can be inherited from other parts or directly imported from other sources.

For each part, main classical methods are implemented but some more original approaches are also offered.

- Various dissimilarity and distance estimations are proposed for different data: quantitative, qualitative, binary, DNA sequence... Properties of dissimilarities are largely explored and transformations are proposed to restore suitable properties when necessary.
- Principal Coordinate analysis that searches for graphical representations on Euclidean plans that conserve at best distances between units.
- Tree construction methods include hierarchical trees with various aggregation criteria (weighted or unweighted), Neighbor-Joining tree (weighted or unweighted), Scores method. Ordinal extensions of NJTree and Scores attempt to reduce sensibility to data error. NJTree under topological constraints allows forcing the a priori known tree structure of some data subsets. Bootstrapping in NJTree and an original method to detect influent units can be used to estimate how the tree is supported by the data.
- Tree representation. Many graphical tools are offered to make graphs easy to read and ready to insert in publication or other document.
- Tree comparisons. When several data sets are used to construct trees on the same unit set, consensus methods, maximum agreement subtree, distances between trees are proposed to compare or synthesize these trees.

This documentation follows the organization of the software menu. Each chapter begins with a brief description of the applied methods or introduces necessary technical features. Input windows, options/parameters and output are then described.

## General features

### ***DARwin file edition***

All the files used by DARwin software are in windows ASCII text format with Tab separator.

These files can be viewed or edited with any Windows text editor. We recommend using of Notepad++ (free software available at <http://notepad-plus-plus.org/>).

This text format is also compatible with spreadsheet programs (Microsoft Excel for example).

## ***Graphical export***

In the 'Tree'-'Draw' and 'Factorial Analysis'-'Graphical representation' windows, it's possible to export and save the drawing windows in EMF (Enhanced Meta File) format.

These EMF files can be imported in vector graphical editors and can be modified, embedded, converted in other formats and more... to insert them in any document like publications or other.

The EMF files are directly compatible with all the Microsoft Office Suite components.

To edit an EMF file in Microsoft PowerPoint software:

- Insert as picture on a slide ;
- Right click on the picture and ungroup the elements as long as the option is purposed (number of groups depends on graph complexity) ;
- Picture external frame can be deleted ;
- All tools are now available depending on type of element (TextBox options for texts, Line options for lines, Shape options for unit symbols...)

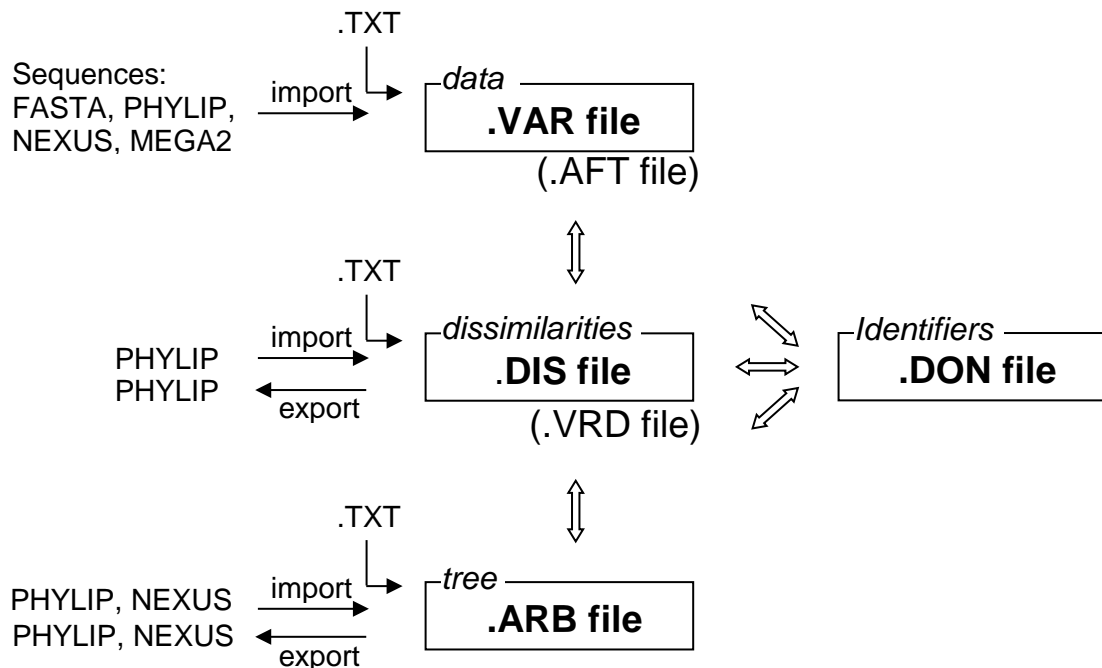
The modified picture can be Re-grouped in a single element (Select all, right click and Group). This element can be exported in different file formats (Right click on picture, Save as Picture...

---



# Data and file formats

DARwin uses its own formats for data files but import/export from/to standard formats in phylogeny are proposed. All files are text ASCII files with Tab as separators; they can be edited with any text editor. They can also be imported or exported directly from spreadsheets like Excel in saving files in text format with Tab as separators. The file structure as described below has to be scrupulously duplicated and the example files provided with the software can be used as models. It is often a convenient way to create or modify data files.



The size of the data set is constrained only by the available physical and virtual memory.




*The algorithmic complexity of a lot of methods is in  $n^2$  or  $n^3$  and even exponential for some ones. Although implemented algorithms are time optimized, some procedures require very long time when the number of units becomes large.*

## General features for file management

The first line in DARwin files is always a signature giving the DARwin version used to create the file (in order to manage compatibility between successive versions) and the type of the file. This header format must be strictly conform (the best is to make a copy/paste from the data examples provided with the software).

A comment field may be found at the end of the file where information on file creation is automatically recorded or updated (cf. sequence data example). This field can be modified by the user with any external editor, [NotePad++](#) for example.

When a file is asked in input,  opens a window giving the main characteristics of the selected file and displaying its comment field.

When a file is asked in output, its extension is automatically generated according to the required type (any other extension is forbidden) and a name is proposed: the same name as the input file if the extension is not the same or the input file name completed with “\_xxx” where xxx is a string specific to each procedure.

Examples:

‘demo.DIS’ for a dissimilarity calculated from ‘demo.VAR’ file

‘sequence\_pruned.ARB’ for a tree resulting of unit pruning in the tree

‘sequence.ARB’

## ***Data file .VAR components***

The format is a classical row/column format with units in row and variables in column. Three types of data can be stored in .VAR files: single data, allelic data and sequence data, the type is given in the header.

The data organization is always the same:

**1st line: header**

**2nd line: number of units - number of variables**

**3rd line: alphanumeric label for each column**

**next lines:** the first field is a numerical **unit identifier**, followed by the **values for each variable**, with Tab as separator between fields.

Unit identifiers are always positive numerical values (see .DON files), they are not necessarily consecutive. However as this value is used as array index in the programme, it is better to keep the greatest value as low as possible and avoid for example identifying a set of four accessions by 1, 10, 100, 1000, that will require tables of dimension 1000 for only 4 values!



*This numerical unit identifier cannot exceed the maximal dimension for integer arrays (32 767).*

### **Single data**

Each unit is characterized by a single value for each variable. This very general format can be used for example in genetics for haploids, for homozygote diploids, for dominant markers...

Variables have continuous or discrete numeric values, including counts, 0/1 data...

Example for 5 units and 4 variables:

@DARwin 5.0 – SINGLE				
5	4			
Unit	Var1	Var2	Var3	Var4
2	610	140	60	10
3	475	90	250	30
9	10	10	495	110
10	615	140	65	999
14	179	29	421	87

## Allelic data

This format is used to record allelic composition for diploids or polyploids, the ploidy  $\pi$  being given in the header. In second line, are given the number of units and the total number of alleles ( $= \pi \times \text{number of loci}$ ). Each unit receives  $\pi$  consecutive values for each of the loci. Allele order has no particular meaning excepted when haplotypes are required. In this case, the alleles of each haplotypes are always in the same order for each locus.

Variables have numeric values identifying the alleles of each locus. The allele code is free; for example, the number of repeats or the string length should be provided as allele code for microsatellite data.

Example for 4 units and 3 loci for a diploid:

@DARwin 5.0 - ALLELIC – 2						
4	6					
N°	M1	M1	M11	M11	M15	M15
1	8	8	1	1	2	2
2	8	1	1	3	1	2
3	8	8	3	3	3	1
7	1	8	3	1	1	2

If phases are known, the alleles of the same haplotype will be recorded at the same position for each locus. For example for unit 2, the first haplotype is 8 1 1 and the second one is 1 3 2.

## Sequence data

Sequence data consist of several aligned sequences of equal length.

In second line, are given the number of sequences and the length of the sequence. The next lines give the sequences with Tab as separator between successive positions. Valid characters are only A,T(U),G,C (upper or lower case) and hyphen (-) for gaps. Any other character is regarded as missing value (including N, X...).

@DARwin 5.0 - SEQUENCE										
5	10									
Unit	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	C	T	T	C	C	A	A	G	C	T
2	C	T	T	N	C	A	A	G	C	-
3	G	t	T	C	C	A	A	G	C	-
4	C	T	T	C	C	A	A	G	C	T
5	C	T	T	C	C	A	a	G	C	T
Imported Sequences...										
from file (in Fasta format) : demo_seq.fas										

## Data file .AFT components

.AFT files have exactly the same structure as [.VAR files](#) and are used to store the coordinates of each unit on selected axes in a factorial analysis (see [Factorial Analysis – Output files](#)). In second line, are given the number of units and the number of retained axes. The next lines give the coordinates of each unit on these axes with Tab as separator.

```
@DARwin 5.0 - AFT
4      3
Unit   CP 1   CP 2   CP 3
1      0.605848642256061 -0.274256439944485 -0.450259969274018
2      0.132039682872075 -0.604884179026046 -7.49942145023929E-02
3      -1.10549598783426 -0.752823165742376 -0.153755560336382
4      0.607204345501523 -0.170470389344589 -0.474296951474667
```

## ***Data file .DIS components***

A DARwin file with extension .DIS stores the dissimilarity lower semi-matrix (without the diagonal) as computed by the software or imported from other formats.

The first line is a fixed header, the second gives the number of units. They are followed by the semi matrix where each unit is identified by its numerical code.

A dissimilarity value has at most 16 digits (including the decimal point), possibly in exponential notation. This high precision on dissimilarity values is required by many numerical algorithms. Of course it results in very large data files for high numbers of units or high numbers of bootstrapped matrices. Negative values are allowed but most of procedures require only non null dissimilarities.

The dissimilarity matrix is followed by a comment field which is automatically filled by the program; it summarizes options retained for dissimilarity computing and inserts the 'comment' component of the .VAR file in input.

```
@DARwin 5.0 - DIS
5
      2      4      10      11
4      0.953652613736441
10     1.88175586432467  1.49913022049294
11     15.2197881575654  0.1    1.04809776643477
12     2.29387992137609E-02  0      1.47521404658095  1.228
Dissimilarity calculated from 'single' data file: Demo.var (type: Continuous)
Dissimilarity index: Usual Euclidean
Missing data options:
no missing data
No Bootstrap
```

In case of bootstrap re-sampling, each semi-matrix is successively recorded at the end of the file. Bootstrapped dissimilarities are recorded in simple precision format (8 digits).

DARwin files with extension .VRD have an identical structure (with VRD instead of DIS in the header). They store the variances associated to dissimilarity values in the companion .DIS file and are used by [MVR Neighbor-Joining](#) tree construction.

## ***Data file .ARB components***

A DARwin file with extension .ARB stores a tree structure described by the list of its edges and their length.

An edge is coded by the labels of the two connected nodes. An external node is identified by the corresponding numerical unit identifier and each internal node receives an increasing number beginning at  $u+1$  where  $u$  is the greatest numerical unit identifier found in the dissimilarity file (even if this unit is not retained in the tree). In general,  $u$  is greater than the number of units, it will be the same only if the  $u$  units are consecutively identified as 1, 2,...,  $u$ .

- **1st line: header**
- **2nd line: number of units in the tree – u value - number of edges**
- **next lines: for each edge, the two connected nodes and its length**

The edge list may be followed by successive structured blocks of complementary information enclosed between 'Tree\_Parameters' and 'End\_Tree\_Parameters' keywords.

A first block concerns bootstrap values (between 'Bootstraps' and 'End\_Bootstraps'). It will appear only in case of tree construction on bootstrapped dissimilarities. The first line gives the number of bootstraps and the number of internal edges. The following lines give the two nodes of an internal edge and its bootstrap value.

The following blocks are optional and keep track of all editing actions on the tree (see [Tree Draw](#)). They allow to redisplay the tree exactly as it was saved. They record different settings like rotation, root... (between 'Settings' and 'End\_Settings'), colors (between 'Colors' and 'End\_Colors'), titles (between 'Titles' and 'End\_Titles')...

A comment field summarizes options retained for tree construction and the dissimilarity comment field is copied.

Example for a tree on 6 units selected in a .DIS file where the greatest unit identifier is 14:

```
@DARwin 6.0 - ARB
```

```
6      14      9
1      18      0.1206
3      17      2.5
4      16      0.0022
6      15      0
8      15      0.0101
9      16      0.0078
15     18      2.7206
16     17      2.5
17     18      0.2206
```

```
Tree_Parameters
```

```
Bootstraps
```

```
100    3
15     18      43
16     17      57
17     18      89
```

```

End_Bootstraps
Settings
0      0      1      1      0      0      0      0      1      0
End_Settings
End_Tree_Parameters
Tree construction: Weighted Neighbor-Joining
calculated from dissimilarity file : test.dis
6 selected units on 14
Bootstraps: 100
Average 'edge' distance between bootstrapped trees: 0.7244
5-percentile: 0.6444
95-percentile: 0.8222
-----
Dissimilarity calculated from 'single' data file: Test.var (type: presence/absence)
Dissimilarity index: Dice
Bootstraps: 100
Missing data options:
no missing data

```



*Tree file format has been modified between version 5 and version 6. However tree files from version 5 are compatible with the version 6 and will be converted while reading.*

## ***Data file .DON components***

In most data formats, unit identifier is a single alphanumeric string (of length 10 in PHYLIP format for example). However the user has often several identifiers for the units (botanical classification, geographical origin...) and he would use them to illustrate and interpret graphical displays. So in DARwin we have retained a double level identification. The first one is referred to as internal identifier, it is a numerical code used in all files (.VAR, .DIS, .ARB) to identify each unit. The second level is stored in a .DON file giving correspondence between the internal identifier and a set of other identifiers, said external identifiers.

The internal identifier is always used by default but the user can always invoke a .DON file and select a more convenient identifier.

A same .DON file may correspond to several different files, it is sufficient that all units of a file have a corresponding entry in the .DON file. For example a tree build on a subset of units may use the initial .DON file that references the whole unit set.

A .DON file is automatically created when dissimilarities or trees are imported from other formats. A sequential numerical value is attributed to each unit and the imported identifier is recorded in a .DON file.

This file is also used to store outputs of some procedures. For example, when user defines clusters in a tree, an identifier is created in a .DON file to store the cluster which the unit belongs to.

The first line is a fixed header, the second line gives the number of units and the number of external identifiers, these external identifiers receives a name in the third line. Following lines give for each unit referenced by its numerical identifier, the list of external identifier values.

At the end a comment field is automatically filled by the programme (for importation for example).

@DARwin 5.0 - DON						
5	6					
N°	Code	Name	Type	Loc.	cl1	cl2
1	B35	Poyo	acu.	China	1	1.0
3	B22	Amer.	acu.	Cam.	1	0.5
4	H3	Blug.	balb.	India	1	0.3
6	H7	Klue	balb.	India	2	0.1
7	B07	Mich.	acu.	Cuba	3	0.2
External Identifiers for file: Example.var						
Imported Sequences...						
from file (in Fasta format): Example.fas						

---

# File menu

## *View File*

This function opens a window displaying the file content according to its type (.VAR, .DIS, .ARB, .DON). Data cannot be modified here.

For .DIS file with bootstrap, a bootstrap navigation frame appears automatically. The complete matrix is initially displayed but any bootstrapped matrix can be selected.

For very large files, e.g. several thousands of units for a dissimilarity file or of variables for a data file, the loading can be rather long. Display is now done by parts. Each part represents 2 hundreds of units horizontally and vertically for dissimilarity file, 2 hundreds of variables horizontally and 2 hundreds of units vertically for data files. A navigation frame appears automatically if needed to select vertical and horizontal range of data and validation is done when pressing the 'View' button.



**Copy** sends current part of data to the window clipboard for copying in any word processor or spreadsheet.



*View File window is multi instantiable and can be opened several times.*

## *Importation / Exportation*

### **Import data matrix**

This function creates a .VAR file from a data matrix in text format with Tab separators, as produced by Excel for example.

The first line of the matrix includes a generic name for the units followed by the identifiers of each variable.

The following lines give the identifier of each unit followed by the value of this unit for each variable. Blanks will be coded as missing data.

For example, the file DataMatrix.TXT:

Unit	a	b	c	d
u1	67	44	67	73
u2	20	87	4	
u3	69	23	80	36
u4	40		77	62
u5	22	31	18	77
u8	24	81	67	19

will give in output the files DataMatrix.VAR and DataMatrix.DON:



@DARwin 5.0 - SINGLE				
6	4			
Unit	a	b	c	d
1	67	44	67	73
2	20	87	4	999
3	69	23	80	36
4	40	999	77	62
5	22	31	18	77
6	24	81	67	19

Imported Single data...

@DARwin 5.0 - DON	
6	1
Unit	Unit
1	u1
2	u2
3	u9
4	u4
5	u5
6	u8

External identifiers for file: datamatrix.var

Imported Single data...

### • File to import

The user has to select the file to import, the default extension is .TXT but any other file extension is allowed.

### • Data type

Select the type **Single** or **Allelic**.

For allelic data, the ploidy  $\pi$  must be specified and each block of  $\pi$  consecutive variables is regarded as a locus. The procedure aborts if the number of variables in the .TXT file is not a multiple of  $\pi$ .

### • Integer code for missing data

Specify the variable value retained to code a missing data in the .VAR file. This value may be found in the initial data matrix and will not be modified. This value will be used to code all missing, non-numeric or bad value found in original file.

### • Save data as...

The output file is necessarily a .VAR file and this extension is automatically added to the file name.

An associated .DON file is created with the same name to record correspondence between the numerical identifier in the .VAR file and the identifier read in the imported file.

## Import sequence

DARwin supports conversions from most common sequence file formats into DARwin format. FASTA, PHYLIP and NEXUS (PAUP) formats are directly converted. For less common formats, we propose to use MEGA2 as intermediate. It is a free software (<http://www.megasoftware.net/>) developed by S. Kumar, K. Tamura, I. Jakobsen and M. Nei, that offers conversion from several formats like CLUSTAL, GCG, PIR, NBRF, MSF, IG, Internet (NCBI) XML in .MEG format. DARwin proposes conversion from .MEG format to import the intermediate files created with MEGA2.



*DARwin allows only hyphen (-) as valid symbol for gaps. Any other character, except A, T(U), G, C (upper or lower case) is recorded but will be regarded as missing value.*

Examples for each imported format are given below.

. Fasta format

```
>Homo
AGTCGAGTC---GCAGAAACGCATGAC-GACCACATTTT-CC
TTGCAAAG

>Pan pani
AGTCGCGTCG--GCAGAAACGCATGACGGACCACATCAT-CC
TTGCAAAG

>Gorilla
AGTCGCGTCG--GCAGATACGCATCACGGAC-ACATCATCCC
TCGAGAG

>Pongo
AGTCGCGTCGAAGCAGA--CGCATGACGGACCACATCATCCC
TTGAGAG
```

. PHYLIP format

PHYLIP interleaved format

```
4      50
Homo      AGTCGAGTC---GCAGAAACGCATGAC
Pan pani  AGTCGCGTCG--GCAGAAACGCATGAC
Gorilla    AGTCGCGTCG--GCAGATACGCATCAC
Pongo      AGTCGCGTCGAAGCAGA--CGCATGAC
-GACCACATTTT-CCTTGCAAAG
GGACCACATCAT-CCTTGCAAAG
GGAC-ACATCATCCCTCGAGAG
GGACCACATCATCCCTTGAGAG
```

PHYLIP non-interleaved (sequential) format

```
4      50
Homo      AGTCGAGTC---GCAGAAACGCATGAC
-GACCACATTTT-CCTTGCAAAG
Pan pani  AGTCGCGTCG--GCAGAAACGCATGAC
GGACCACATCAT-CCTTGCAAAG
Gorilla    AGTCGCGTCG--GCAGATACGCATCAC
GGAC-ACATCATCCCTCGAGAG
Pongo      AGTCGCGTCGAAGCAGA--CGCATGAC
GGACCACATCATCCCTTGAG
```

. NEXUS format

```
#nexus
begin data;
dimensions ntax=4 nchar=50;
format datatype=dna interleave=yes gap=- missing=?;
matrix
Homo      AGTCGAGTC---GCAGAAACGCATGAC
Pan pani  AGTCGCGTCG--GCAGAAACGCATGAC
Gorilla    AGTCGCGTCG--GCAGATACGCATCAC
Pongo      AGTCGCGTCGAAGCAGA--CGCATGAC

Homo      -GACCACATTTT-CCTTGCAAAG
Pan pani  GGACCACATCAT-CCTTGCAAAG
Gorilla    GGAC-ACATCATCCCTCGAGAG
Pongo      GGACCACATCATCCCTTGAGAG
;
end
```

. MEGA format

```
#mega
TITLE : DNA sequence data - test
#Homo
AGTCGAGTC---GCAGAAACGCATGAC-GACCACATTTT-CCTTGCAAAG
#Pan pani
AGTCGCGTCG--GCAGAAACGCATGACGGACCACATCAT-CCTTGCAAAG
#Gorilla
AGTCGCGTCG--GCAGATACGCATCACGGAC-ACATCATCCCTCGAGAG
#Pongo
AGTCGCGTCGAAGCAGA--CGCATGACGGACCACATCATCCCTTGAGAG
```

### • File to import

The user has to select the file to import, he can select a file type (.FAS, .PHY, .PAU, .MEG) but any other file extension is allowed.

If the file extension is a known extension (.FAS, .PHY, .PAU, .MEG), the file will be assumed to be in this format (e.g. Fatsa for .FAS) but another **import format** can be specified: FASTA, PHYLIP, PAUP(NEXUS), MEGA.

### • Save sequences as...

DARwin creates a .VAR sequence file where each unit receives a consecutive numerical identifier. All blanks in sequences are removed.

An associated .DON file is created with the same name to record correspondence between the numerical identifier in the .VAR file and the identifier read in the imported file.

```
@DARwin 5.0 - SEQUENCE
4      50
Unit   P1      P2      P3      P4      P5      P6      P7      P8      P9      ... P50
1      A      G      T      C      G      A      G      T      C      ... G
2      A      G      T      C      G      C      G      T      C      ... G
3      A      G      T      C      G      C      G      T      C      ... G
4      A      G      T      C      G      C      G      T      C      ... G
Imported Sequences...
from file (in Fasta format): demo_seq.fas
```

```
@DARwin 5.0 - DON
4      50
Unit   Name
1      Homo
2      Pan pani
3      Gorilla
4      Pongo
External Identifiers for file: seq_fast.var
Imported Sequences...
from file (in Fasta format): demo_seq.fas
```

The procedure aborts if the file does not correspond to the specified format or if sequences have not the same length.

N.B. In MEGA files, the point symbol is allowed and codes for the same symbol as in the first sequence. DARwin replaces the point by the symbol met in the first sequence.



*Some formats allow recording several data sets in a same file (for example, SEQBOOT procedure in PHYLIP) but DARwin will import only the first set.*

## Import dissimilarity

DARwin supports conversions from NTSYS format and PHYLIP format which is the most common format and is read by all major software.

PHYLIP files characteristics:

- UNIX or Dos text file format
- [Tab] separator between each field
- Lower semi-matrix or Full-matrix

#### Example for semi-matrix PHYLIP file:

```

10
U68591
U68664 0.394891
U68592 0.414843 0.442027
U68686 0.371826 0.414843 0.276453
U68609 0.490123 0.521929 0.416497 0.364001
U68638 0.363032 0.389033 0.268092 0.061001 0.376207
U68643 0.518779 0.583528 0.556294 0.485847 0.316086 0.474782
U68659 0.422482 0.514325 0.277780 0.114629 0.393321 0.095229 0.505483
U68627 0.500022 0.523433 0.420349 0.371596 0.054455 0.360652 0.340769
      0.395244
U68633 0.395174 0.428856 0.298352 0.159261 0.365873 0.169421 0.422792
      0.165149 0.377522

```

#### Example for full-matrix PHYLIP file:

```

10
U68589 0.000000 0.337144 0.360977 0.415506 0.287299 0.297057 0.392240
      0.309315 0.320066 0.328638
U68590 0.337144 0.000000 0.378254 0.319757 0.169021 0.329311 0.273158
      0.312653 0.266838 0.206259
U68591 0.360977 0.378254 0.000000 0.414843 0.336162 0.356376 0.427517
      0.322673 0.352060 0.344952
U68592 0.415506 0.319757 0.414843 0.000000 0.284235 0.332574 0.229894
      0.363330 0.325227 0.265168
U68593 0.287299 0.169021 0.336162 0.284235 0.000000 0.276866 0.283055
      0.291774 0.217362 0.189372
U68594 0.297057 0.329311 0.356376 0.332574 0.276866 0.000000 0.364319
      0.280537 0.263379 0.251328
U68595 0.392240 0.273158 0.427517 0.229894 0.283055 0.364319 0.000000
      0.360148 0.317196 0.281804
U68596 0.309315 0.312653 0.322673 0.363330 0.291774 0.280537 0.360148
      0.000000 0.276011 0.259990
U68597 0.320066 0.266838 0.352060 0.325227 0.217362 0.263379 0.317196
      0.276011 0.000000 0.213189
U68598 0.328638 0.206259 0.344952 0.265168 0.189372 0.251328 0.281804
      0.259990 0.213189 0.000000

```

#### • File to import

The user has to select the file to import, he can select a file type (.PHY or other) but any other file extension is allowed.

If the file extension is a known extension (.PHY), the file will be assumed to be in this format (e.g. PHYLIP for .PHY) but the correct import format can be specified: PHYLIP or NTSYS.

#### • Save dissimilarity as...

The output file is necessarily a .DIS file and this extension is automatically added to the file name.

An associated .DON file is created with the same name to record correspondence between the numerical identifier and the identifier read in the imported file.

```

@DARwin 5.0 - DIS
5
      1      2      3      4
2      0.382632
3      0.374527      0.387518
4      0.366292      0.456798      0.401701
5      0.382632      0.238441      0.414975      0.443476
Imported dissimilarities...
from file (in PHYLIP format): dist_phy.phy

```

```
@DARwin 5.0 - DON
5      1
Unit   Name
1      Cow
2      Carp
3      Chicken
4      Human
5      Loach
External Identifiers for file: seq.var
from file (in PHYLIP format): dist_phy.phy
```



*Several dissimilarity matrices can be recorded successively in a same PHYLIP file but DARwin will import only the first matrix.*

## Export dissimilarity

DARwin supports conversions from .DIS format into NTSYS format and PHYLIP format which is a standard format read by all major software.

### • Dissimilarity

The input file is necessarily a .DIS file.



Open a window summarizing main characteristics of the selected dissimilarity.

### • Identifiers

An external identifier can be selected in an associated .DON file. This identifier will be used as unit label in exported file. If no identifier is chosen, the numerical identifier read in the .DIS file will be used as unit identifier in the output file.



*PHYLIP identifiers have at most 10 characters and any longer DARwin identifier will be truncated to the 10 first characters in the exported file.*  
*- All not valid symbols in PHYLIP identifiers will be translated in a valid symbol: accented vowel in non-accented vowel ...*

### • Save dissimilarity as...

The user has to give a name for the exported file, he can select a file type (.PHY or other) but any other file extension is allowed.

If the file extension is a known extension (.PHY), the file will be assumed to be in this format (e.g. PHYLIP for .PHY) but another export format can be specified: PHYLIP or NTSYS.



*DARwin records bootstrapped dissimilarity matrices after the main matrix in the same .DIS file but only the main matrix will be exported.*

## Export dissimilarity as column

DARwin can export from .DIS format into text column file format.

### • Dissimilarity

The input file is necessarily a .DIS file.



Open a window summarizing main characteristics of the selected dissimilarity.

- **Save dissimilarity as...**

The user has to give a name for the exported file, he can select a file type (.txt or other) but any other file extension is allowed.



*DARwin records bootstrapped dissimilarity matrices after the main matrix in the same .DIS file but only the main matrix will be exported.*

Example of dissimilarity export as column:

i	j	D(i,j)
2	1	0.09375
3	1	0.09375
3	2	0.0625
4	1	0.1875
4	2	0.09375
4	3	0.09375
5	1	0.125
5	2	0.09375
5	3	0.0625
5	4	0.125
		...

## Import tree

PHYLIP and NEXUS are very common treefile formats used by a lot of major programs. They are based on the Newick Standard that uses a system of nested parentheses for coding a tree.

PHYLIP tree format is a strict Newick format:

```
(Bovine,(Gibbon,(Orang,(Gorilla,(Chimp, Human)))) , Mouse);
```

or when branch lengths are specified:

```
(Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,(Gorilla:0.17147,
(Chimp:0.19268,Human:0.11927):0.08386):0.06124):0.15057):0.54939,
Mouse:1.21460);
```

or with bootstrap values:

```
(Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,(Gorilla:0.17147,
(Chimp:0.19268,Human:0.11927)0.75:0.08386)0.80:0.06124)0.45:0.15057)
0.23:0.54939, Mouse:1.21460)0.89;
```

(For consensus from bootstrapped trees, PHYLIP gives the bootstrap values as branch lengths.)

NEXUS format adds to the Newick tree description some commands relevant to other programs such as PAUP:

```
#NEXUS
Begin trees;
Tree PAUP_1 = [&U] (Bovine:0.69395,(Gibbon:0.36079,(Orang:0.33636,
(Gorilla:0.17147,(Chimp:0.19268, Human:0.11927):0.08386):0.06124)
:0.15057):0.54939,Mouse:1.21460);
end;
```

### • File to import

The user has to select the file to import, he can select a file type (.PHY, .TRE or other) but any other file extension is allowed.

If the file extension is a known extension (.PHY or .TRE), the file will be assumed to be in this format (PHYLIP for .PHY and NEXUS for .TRE) but another **import format** can be specified: PHYLIP or NEXUS.

### • Save tree as...

The user has to give a name for the output file. It is necessarily a .ARB file and this extension is automatically added to the file name.

An associated .DON file is created with the same name to record correspondence between the numerical identifier and the identifier read in the imported file.



*Several trees can be successively recorded in a same PHYLIP or NEXUS file, DARwin creates as many .ARB files as trees in the imported file. If filename is the name given by the user for the output file, these files will be labelled filename\_1.ARB, filename\_2.ARB ...*



*If branch lengths are not specified in the imported tree, they will be set to 1 in the DARwin file.*

## Export tree

DARwin supports conversions from .ARB tree format into PHYLIP and NEXUS formats that are standard formats read by all major software.

### • Tree to export

The input file is necessarily a .ARB file.

### • Tree parameters

Open the [tree display window](#) to define tree parameters required by the export format:

- Newick tree format is not invariant with the root position. The root retained for exportation will be the current root used to draw the tree, it can be modified with the appropriate tool.
- Newick format requires a single unit identifier. This identifier will be the current identifier used for the tree display, it can be the numerical identifier as read in the .ARB file or any external identifier selected in the associated .DON file.



*PHYLIP identifiers have at most 10 characters and any longer DARwin identifier will be truncated to the 10 first characters in the exported file.*

*All not valid symbols in PHYLIP identifiers will be translated in a valid symbol: accented vowel in non-accented vowel...*

- **Save tree as...**

The user has to give a name for the exported file, he can select a file type (.PHY or .TRE) but any other file extension is allowed.

If the file extension is a known extension (.PHY or .TRE), the file will be assumed to be in this format (PHYLIP for .PHY or NEXUS for .TRE) but another **export format** can be specified: PHYLIP or NEXUS.

---



# Dissimilarity menu

## *Method*

The most general mathematical object representing the difference between two units of a set  $E$  is a **dissimilarity**. It is a function  $d$  of the set of pairs  $(i, j)$  of  $E \times E$  in the positive or null real numbers, which is symmetrical ( $d(i, j) = d(j, i)$ ) and such that  $d(i, i) = 0$  for any  $i$ . This definition is relatively simple and covers a large number of possible measures but, on the other hand, opens up only a few mathematical properties.

It is often necessary to opt for more constrained definitions when particular mathematical properties are required by the applied methods.

This general definition allows some incoherencies since for two identical units ( $d(i, j) = 0$ ) it is possible that for some unit  $k$ ,  $d(i, k) \neq d(j, k)$ . So we often add the condition:  $d(i, j) = 0 \Rightarrow d(i, k) = d(j, k)$  to obtain an **even dissimilarity** (or semi-proper dissimilarity).

A **distance**, called also a **metric**, is a particular even dissimilarity obtained by adding the condition  $d(i, j) = 0 \Rightarrow i = j$  and the triangular inequality between three units:  $d(i, j) \leq d(i, k) + d(j, k)$ . This natural condition translates simply into the possibility of representing any triplet of points by a two-dimensional triangle. This very useful property avoids the problem of negative edge lengths in agglomerative tree construction methods. An important group of distances is made up of Minkowski distances of order  $p$  ( $p \geq 1$ ). A distance belongs to this group if it can be found an integer  $K$  and a series of  $K$  values  $x_{ik}$  for each unit  $i$ , the distance thus being written as:

$$d(i, j) = \left( \sum_K |x(i, k) - x(j, k)|^p \right)^{1/p}$$

The only cases used in practice correspond to the values 1 and 2 for  $p$ . When  $p = 1$ , the index is known as the Manhattan distance (or City-Block distance or L1 distance):

$$d(i, j) = \sum_K |x(i, k) - x(j, k)|$$

The usual Euclidean distance corresponds to  $p = 2$ :

$$d(i, j) = \sqrt{\sum_K (x(i, k) - x(j, k))^2}$$

These two distances have some interesting properties for geometrical representation, and Euclidean distance for example is required for factorial analysis.

Of course, Manhattan or Euclidean distances are obtained when the distance is calculated from a table of units  $\times$  variables by summing, the absolute values or the squares of the deviations between  $i$  and  $j$ . However, some indices, for example those calculated on presence/absence data, can be of either of these two types without this following evidently from the mode of construction.

Manhattan and Euclidean distances belong to the same distance family and are linked. It can be shown, for example, that any Euclidean distance is a Manhattan distance and that the square root of a Manhattan distance is a Euclidean distance.

Finally, some very constrained distances have the property that they can be represented as the path length between leaves in a tree. Tree construction methods all aim to approximate as closely as possible the observed dissimilarity by one of these particular distances.

The best known of these distances is the ultrametric which satisfies the ultrametric condition: for any three units  $i$ ,  $j$  and  $k$ :

$$d(i, j) \leq \max(d(i, k), d(j, k))$$

This condition expresses that among the three distances between three points, the two largest are equal. Any triplet of points thus forms a sharp isosceles triangle. This property allows a representation in the usual form of a dendrogram, which is a tree that has a particular point, the root, located at an equal distance from all the leaves of the tree.

In the 1970s, the distance called the additive tree distance was proposed. It verifies the four points condition:

$$d(i, j) + d(k, l) \leq \max(d(i, k) + d(j, l), d(i, l) + d(j, k))$$

which expresses that for any four individuals  $i$ ,  $j$ ,  $k$  and  $l$ , among the three sums of distances two by two, the two largest are equal. This property allows a tree representation where leaves are not constrained to be at an equal distance from a root as for ultrametrics. This lesser constraint thus enables a more faithful representation of initial dissimilarities. The ultrametric condition is shown to be only a particular case of the four points condition.

## ***Dissimilarity estimation***

A dissimilarity matrix is computed from unit by variable data read in a .VAR file and is stored in a .DIS file. Specific windows are opened according to the data type (single, allelic or sequence).

## Common options

### Input / output files

A .VAR file is selected as input. The reading procedure aborts if the type read in the file header does not correspond to the expected type.

A name for the output .DIS file has to be chosen. By default, DARwin proposes to keep the same name as the input file with the .DIS extension.

### Missing data

Dissimilarity between two units can be evaluated only for variables with valid values for the two units, so any invalid value (missing data) has to be discarded for the dissimilarity calculation.

Except for Sequence data, the variable value retained to code a missing data in the .VAR file has to be specified by the user (**Integer code for missing data**).

Several options are proposed to discard missing data:

- **Complete unit deletion**

Discard any unit with at least one missing data, the dissimilarity matrix will contain only units without missing data.

- **Complete variable deletion**

Discard any variable with at least one missing data (this variable is discarded for all units). The dissimilarity matrix will contain all the units but the values will be calculated only on variables without any missing data.

- **Pairwise deletion (option by default)**

If for a pair, at least one of the two values is missing for a variable, this variable is discarded for computing dissimilarity between these units (and only for them). The dissimilarity matrix will contain all the units but the values will be calculated on a number of variables that may change for each pair.

A minimal proportion of valid variables for a pair can be chosen (50, 60, 70, 80 or 90%). For the first pair which does not reach this threshold, the procedure aborts and an information window identifies the incriminate pair.

The two first options discard all missing data from the data set. They should be used when missing data are concentrated in some units or variables only. If missing data are distributed more or less at random, a great number of valid data will be also discarded with complete deletion options. Pairwise deletion option avoids this loss of information in removing only missing data for the considered pair.



*For pairwise deletion, dissimilarities between units will not be estimated from the same variable set. This may induce some troubles in dissimilarities properties.*

	...P...Q...
i	...A...N...
j	...N...G...
k	...A...T...

For example, let  $i, j, k$  be three DNA sequences of length  $n$  that are exactly the same except for two positions  $p$  and  $q$  (N being a missing value).

For a dissimilarity  $d$  defined as the unmatching number,  
 $d(i, j)=0$  since the two sequences are identical on the  $n-2$  retained positions,  
 $d(i, k)=0$  since the two sequences are identical on the  $n-1$  retained positions,  
 $d(j, k)=1$  since the two sequences differ for one position on the  $n-1$  retained positions.

So  $d(i, j)=0$  but  $d(i, k) \neq d(j, k)$  and the even property is not verified although the dissimilarity is theoretically an even dissimilarity.

## Unit selection

This option allows calculating dissimilarity matrix on a user defined subset of units.

The currently unselected/selected units are listed in the left/right columns.



To move part or all units between columns

- **Identifiers** to select a .DON file and an identifier in this .DON file if selection is easier using another unit identifier. Selected and unselected list can be sorted by click on columns header.

. **Statistics on current selection** open a window listing some synthetic parameters for each unit.



**Print** the text on the current printer



**Record** the window as .TXT or .RTF file

## Variable selection

This option allows calculating dissimilarity matrix on a user defined subset of variables.

The currently unselected/selected variables are listed in the left/right columns.



To move part or all variables between columns

. **Statistics on current selection** open a window listing some synthetic parameters for each variable.



**Print** the text on the current printer



**Record** the window as .TXT or .RTF file

## Record selected subset

If checked, this option creates a new .VAR file recording a data matrix with only units and variables selected by the user. The unit numerical identifiers are those

of the initial .VAR file and any associated .DON file stays operational. By default, the proposed filename is name\_sel.VAR where name is the initial .VAR file name. This option is very useful when a same tedious selection has to be used for several purposes.

## Bootstrap

Bootstrap analysis creates a series of bootstrap samples of the same size as the original data: same units on a variable set of same size but in drawing at random and with replacement each variable in the initial variable set. So each variable in the initial dataset may appear 0, 1, 2, 3 ... times in a bootstrap sample. Dissimilarity matrices are computed for bootstrap samples with the same options as for initial matrix, they are successively stored at the end of the .DIS file.

In case of missing data, random sampling may generate units with many missing data. If the number of missing data for a pair in the initial data exceeds a fixed threshold, a warning window points out to the user that dissimilarities might be estimated in bootstraps on a small number of variables for this pair. The threshold for missing data is equal to 0.8 times the threshold chosen for initial data.

The user has only to choose the **number of bootstraps**.



**New in DARwin 6:** The bootstrap iterations are distributed to exploit multi cores processors. A textbox informs on computing steps; a main progress bar indicates global progression and secondary progress bars indicates progression on the different cores (only for the 4 first cores). A 'Cancel' button is available to abort the procedure.

## Dissimilarity for Single data

Data of different nature can be stored as 'Single' data and only a specific family of dissimilarity indices is relevant for each one. So the user has to specify what kind of data is used: **Continuous, Counts, Modalities, Presence/Absence**. The list of available dissimilarity indices is contextually adapted according to this specification.

The procedure is not able to verify that the data correspond to the user specification, except for presence/absence where the procedure aborts if it finds a value not equal to one of the three codes defined for presence, absence or missing data.

### Input / output files

(See above [Input / output files](#) in Common options)

### Missing data

(See above [Missing data](#) in Common options)

## Unit selection / variable selection

(See above [Unit selection](#), [Variable selection](#) and [Record selected subset](#) in Common options)

The **Statistics on current selection** window summarizes for the current selected data set and for each unit (or variable) in the selection:

- number of missing data (according to the missing code defined by the user)
- minimal values and numbers of minimal values
- maximal values and numbers of maximal values
- total number of minimal and maximal values
- mean on valid values for the unit (or the variable)

## Bootstrap

(See above [Bootstrap](#) in Common options)

## Dissimilarity indices – Continuous

Euclidean and Manhattan distances are commonly used for continuous data. The square in Euclidean distances emphasizes on large differences comparatively to small differences. Manhattan distance has a more neutral point of view.

These distances are often divided by the number of variables, thus distance values can be compared between data sets with different numbers of variables.

Manhattan distances can be reduced by the range (max – min) to keep variations in [0,1].

However the range depends only on two particular values of the distribution and these extremes may correspond to very particular units, to some abnormal measures or possibly to errors... We would prefer a criterion based on the whole data set (like the standard deviation). If a Gaussian distribution can be assumed for the variables, an expected range can be deduced from the standard deviation:

$$ER_n = \delta_n \sigma$$

where  $\delta_n$  depends on the number of units in the data set.

If differences greater than the expected range are met, the contribution of the variable is truncated to 1.

notations:

$d_{ij}$ : dissimilarity between units  $i$  and  $j$

$x_{ik}, x_{jk}$ : values of variable  $k$  for units  $i$  and  $j$

$x_{i.}, x_{j.}, x_{.k}$ : mean for units  $i$  and  $j$  or variable  $k$

$x_{..}$ : overall mean

$\max_k - \min_k$ : range for variable  $k$

$ER_n$ : expected range for a set of  $n$  units

$K$ : number of variables

- Usual Euclidean

$$d_{ij} = \sqrt{\sum_1^K (x_{ik} - x_{jk})^2}$$

- Mean Euclidean

$$d_{ij} = \sqrt{\frac{1}{K} \sum_1^K (x_{ik} - x_{jk})^2}$$

- Manhattan (=City-Block, = L1)

$$d_{ij} = \frac{1}{K} \sum_1^K |x_{ik} - x_{jk}|$$

- 'range' Manhattan

$$d_{ij} = \frac{1}{K} \sum_1^K \frac{|x_{ik} - x_{jk}|}{\max_k - \min_k}$$

- 'Expected range' Manhattan

$$d_{ij} = \frac{1}{K} \sum_1^K \min\left(\frac{|x_{ik} - x_{jk}|}{ER_n}, 1\right)$$

**Standardized** option divides each  $x_{ik}$  by the standard deviation of the variable  $k$  calculated on the selected units.

### Dissimilarity indices - Count

This measure expresses a value  $x_{ik}$  as its contribution to the sum  $x_{i.}$  on all variables and is a comparison of unit profiles.

- Chi-2

$$d_{ij} = \sqrt{\sum_1^K \frac{x_{i.}}{x_{.k}} \left( \frac{x_{ik}}{x_{i.}} - \frac{x_{jk}}{x_{.k}} \right)^2}$$

### Dissimilarity indices – Modalities

These indices concern categorical data with several unordered modalities including 0/1 data where 0 and 1 are only particular codes for binary variables with two modalities. (0 and 1 codes could be exchange without consequence on dissimilarity measures, see below for true 0/1 data like presence/absence data).

notations:

$d_{ij}$ : dissimilarity between units  $i$  and  $j$

$u$ : number of unmatching variables

$m$ : number of matching variables

- Rogers & Tanimoto

$$d_{ij} = \frac{2u}{m + 2u}$$

- Sokal & Michener (=simple matching)

$$d_{ij} = \frac{u}{m + u}$$

- Sokal & Sneath (un1)

$$d_{ij} = \frac{u}{2m + u}$$

## Dissimilarity indices – Presence/Absence

These indices concern 'presence/absence' data where only 'presence' modality is informative, modality 'absence' expressing mainly an absence of information. These two modalities are not symmetrical and their exchange leads to a completely different dissimilarity value. All these indices consider that a common absence for two units is no informative to measure their dissimilarity.



*For 0/1 data, it is not always easy to decide if we have really 'presence/absence' data as defined above or only categorical data with only two modalities for which code in 0/1 is purely arbitrary (and can be exchanged without consequence on dissimilarity values) and for which indices for modalities have to be used. Concerning genetic markers of diversity, the decision will be made on biological knowledge (dominant / co dominant, homo / heterozygotes...). For instance RFLP markers read as 0/1 data (presence or absence of a band) are true presence / absence data but dominant AFLP markers will be regarded as categorical data.*

Usually 'presence' is coded as 1 and 'absence' as 0 but the user can specify any other coding values.

notations:

$d_{ij}$ : dissimilarity between units  $i$  and  $j$

$x_i, x_j$ : variable values for units  $i$  and  $j$

$a$ : number of variables where  $x_i = \text{presence}$  and  $x_j = \text{presence}$

$b$ : number of variables where  $x_i = \text{presence}$  and  $x_j = \text{absence}$

$c$ : number of variables where  $x_i = \text{absence}$  and  $x_j = \text{presence}$

- Dice

$$d_{ij} = \frac{b + c}{2a + (b + c)}$$

- Ochiai

$$d_{ij} = 1 - \frac{a}{\sqrt{(a + b)(a + c)}}$$

- Jaccard

$$d_{ij} = \frac{b + c}{a + (b + c)}$$

- Sokal & Sneath (un2)

$$d_{ij} = \frac{2(b + c)}{a + 2(b + c)}$$

## Dissimilarity for Allelic data

### Input / output files

(See above [Input / output files](#) in Common options)

The ploidy  $\pi$  is read in the .VAR header.



## Missing data

(See above [Missing data](#) in Common options)



*Code for missing value concerns allele values but complete or pairwise deletion does not consider each allele value but the blocks of the  $\pi$  alleles of a same locus. A locus is missing for a unit as soon as one of its  $\pi$  alleles is missing.*

## Unit selection

(See above [Unit selection](#), [Variable selection](#) and [Record selected subset](#) in Common options)

The **Statistics on current selection** window summarizes information on selected units or loci, on missing data and gives for each unit:

- the number of missing alleles and missing loci (at least one allele missing)
- the number of homozygote loci
- the number of heterozygote loci

## Locus selection

(See above [Variable selection](#) in Common options)

In loci selection window, all alleles are listed but selection operates on blocks of  $\pi$  alleles of a same locus.

The **Statistics on current selection** window summarizes information on selected units or loci, on missing data and gives for each locus:

- the number of missing alleles
- the number of alleles (= the number of different values found)
- the smallest and the greatest allele code

## Bootstrap

(See above [Bootstrap](#) in Common options)



*The bootstrap procedure samples at random in the set of loci (blocks of  $\pi$  alleles) and not in the set of individual alleles.*

## Dissimilarity indices

notations:

$d_{ij}$  : dissimilarity between units  $i$  and  $j$   
 $L$  : number of loci  
 $\pi$  : ploidy  
 $m_l$  : number of matching alleles for locus  $l$

ex: for  $\pi = 2$

$i$	11	12	12	11	11	12	12	11	11	12
$j$	11	12	21	12	21	23	32	22	23	34
$m_l$	2	2	2	1	1	1	1	0	0	0

- Simple matching

$$d_{ij} = 1 - \frac{1}{L} \sum_{l=1}^L \frac{m_l}{\pi}$$

## Dissimilarity for Sequence data

Dissimilarity between two sequences is usually estimated as the proportion of unmatching positions between them, assuming that multiple nucleotide substitutions are rare and that unmatching count gives an accurate estimation of the mutation number. However this assumption does not hold for long divergence times since frequencies of multiple substitutions at a same position are no longer negligible. These multiple substitutions are hidden and it is not possible to evaluate their numbers from the data themselves. However, a model of evolution being given, it is possible to deduce a statistical estimation. Assuming that all substitutions at a given site are rare, independent, equally probable events, Jukes and Cantor (1969) propose to adjust the substitution number per time unit at a given site by a Poisson distribution. Then assuming that all sites are equivalent, they derive a distance correction for multiple substitutions. Moreover, using the variance of a Poisson distribution, the variance of the corrected distance can be estimated. These variances will be used in tree construction to take into account the precision of the distance estimations (see [MVR – Neighbor-Joining](#)).

Latter, in relaxing some assumptions of the Jukes and Cantor evolution model, several other corrections were proposed in the literature. DARwin includes only simple models requiring few parameters since for more sophisticated models, parameter estimations are rarely available.

notations:

$d_{ij}$  : dissimilarity between sequences  $i$  and  $j$   
 $u_s$  : number of transitions  
 $u_v$  : number of transversions  
 $u = u_s + u_v$  : total number of unmatching sites  
 $m$  : number of matching sites  
 $L = u + m$  : number of valid sites  
 $u_g$  : number of unmatching gaps (or gap blocks)  
 $m_g$  : number of matching gaps (or gap blocks)  
 $a$  : gamma parameter

## Input / output files

(See above [Input / output files](#) in Common options)

If Variances is checked, the variances of the dissimilarities will be calculated and recorded in a file with the same name as the output .DIS file but with extension .VRD.

## Unit selection / site selection

(See above [Unit selection](#), [Variable selection](#) and [Record selected subset](#) in Common options)

Like in other modules, **Unit selection** or **Site selection** allow the user to define units and/or sites which have to be discarded for computing dissimilarities.

In each case, the button **Statistics on current selection** opens a window summarizing information on numbers of selected units or sites, on gaps, on missing data and gives for each selected unit or site, the numbers and the frequencies of gaps, missing data, valid values, A, T/U, G and C.

### Codon position

If sequences correspond to coding part of the genome, dissimilarities can be calculated on user-defined positions to take into account differences in selection pressure on codon positions.

**All** is used for non-coding sequences or if the three positions are kept.

**1st 2nd 3rd**: dissimilarities are calculated only for selected positions, at least one and at most two (three positions being equivalent to **All**).

Obviously sequences must be continuous strings in the input file and a codon position code (1, 2 or 3) is affected sequentially to successive sites of the sequence, the **position of the first site** in the codon being given as parameter.

### Missing data

(See above [Missing data](#) in Common options)

For sequence data, the missing code has not to be user defined. Valid characters are only A, T(U), G, C and hyphen (-) for gaps. Any other character is regarded as missing value (including N, X...).



*Missing data may be frequent in sequence data, particularly if gaps are toggle to missing. When Pairwise site deletion option is selected, numbers of valid sites vary between pairs and if these numbers are too different, dissimilarity comparison becomes questionable.*

### Gaps

Gaps required for multiple alignment reveal insertion-deletion mutational events. A first approach is to consider that these mutational events cannot be compared to substitution events and so cannot be retained to evaluate dissimilarities between units. Then gaps are regarded as missing data and follow in that the options defined for missing data (see above).

Another approach is to consider that indel events account for part of the divergence between two units and have to contribute to the dissimilarity estimation.

Then a gap could be seen as a fifth element. However corrections for multiple substitutions (see below) are not necessary since multiple indel events at the same position seems improbable. DARwin proposes to estimate dissimilarity as the weighted sum of two terms, the first one accounting for substitutions and its correction for multiple substitutions, the second one accounting for deletions and insertions.

For example for a Jukes & Cantor multiple substitution correction (see above for notation):

$$d_{ij} = \left( \frac{m+u}{m+u+m_g+u_g} \right) \left[ -\frac{3}{4} \ln \left( 1 - \frac{4}{3} \frac{u}{L} \right) \right] + \left( \frac{m_g+u_g}{m+u+m_g+u_g} \right) \left[ \frac{u_g}{m_g+u_g} \right]$$

- the first term between [] is the Jukes & Cantor correction for valid sites
- the second term is the unmatching index for gaps
- the two terms between brackets are weights corresponding to part of valid sites and gaps in the total number of sites.

More generally, this equation can be rewritten

$$d_{ij} = f X + g Y$$

where X and Y concern substitution and insertion/deletion events, with f and g the corresponding weights.

The variance of  $d_{ij}$  is the variance of a sum of two terms and is function of the variances of each term but also of their covariance. The frequencies of substitutions and indels are very probably linked, the two terms cannot be regarded as independent but their covariance is not known. We will assume that the two terms are completely positively linked and the covariance will be taken as the product of the two variance square roots. The variance will be so over estimated but in any case under estimated.

Component  $V(Y)$  is estimated assuming a Poisson model for insertion/deletion events. Component  $V(X)$  is the variance of substitution events depending on the retained model, for example for Jukes and Cantor model:

$$V(X) = \frac{u}{L} \left( 1 - \frac{u}{L} \right) \left/ \left( 1 - \frac{4}{3} \frac{u}{L} \right)^2 \right. L$$

Then the variance of  $d_{ij}$  is calculated as:

$$V_{ij} = \left( f \sqrt{V(X)} + g \sqrt{(m_g u_g) / (m_g + u_g)^2} \right)^2$$

The previous formula for dissimilarity considers that each site with a gap corresponds to an indel event (single gap correction). However indel events concern often blocks of consecutive sites and a row of adjacent gaps has to be treated as a single gap, regardless of its length (gap block correction). Then  $u_g$  and  $m_g$  in the previous equation are numbers of unmatching and matching gap blocks.

Optionally, a **minimal length for gap blocks** can be specified.

AGTAC-GTCA
AGTACCGTCA
AGTAC-GTCA
AGTAC-GTCA
AGTAC-GTCA

A frequent error in sequencing is the accidental duplication of a site. This leads to a site with gap inserted for all units except one. In giving 2 as minimal value for block length, only blocks of two sites or more will contribute to the dissimilarity and single gaps (blocks of length 1) will be regarded as missing data and deleted.

The gap option sub-window proposes:

- **Gaps as missing data**

Regard gaps as missing data and follows options defined for missing data.

- **Pairwise single gap correction**

Compute a correction term from each site with at least one gap for pair  $i$  and  $j$ .

- **Pairwise gap blocks correction**

Compute a correction term from each block of consecutive gaps for pair  $i$  and  $j$ .

**Minimal length for gap blocks** gives the minimal number of consecutive gaps to form a block.

(N.B. a missing data in a gap block does not interrupt the block)

Ex:

```
unit i:  A C T C A - - - A A G - C T G A G T C - - G T C
unit j:  T A T G A G T A A G G - T G G - G T C - - T T C
```

Gaps as missing data:  $m_g = 0$   $u_g = 0$

Pairwise single gap correction:  $m_g = 3$   $u_g = 4$

Pairwise gap block correction, 1 as minimal block size:  $m_g = 2$   $u_g = 2$

Pairwise gap block correction, 2 as minimal block size:  $m_g = 1$   $u_g = 1$



*Compatibility between selection options*

*User selection, codon position restriction, missing data, gaps, may result in site or unit deletions. These options are cumulative and a site or a unit is deleted if it is given as removable at least one time.*

Example:

- user deletion of sites 6, 7, 8
- complete site deletion for missing data
- restriction to codon positions 2 and 3 for strings beginning in position 2

site	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	A	T	C	N	N	A	A	C	N	G	X	T	T	A	G
codon position	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1
user selection	+	+	+	+	+	-	-	-	+	+	+	+	+	+	+
codon selection	+	+	-	+	+	-	+	+	-	+	+	-	+	+	-
missing data	+	+	+	-	-	+	+	+	-	+	-	+	+	+	+
resulting selection	+	+	-	-	-	-	-	-	-	+	-	-	+	+	-

(+/- selected/unselected sites)

## Bootstrap

(See above [Bootstrap](#) in Common options)

All options for codon positions, missing data, gaps are maintained for bootstrap resampling.

## Dissimilarity indices

- No correction for multiple substitutions

*simple matching*

$$d_{ij} = \frac{u}{L}$$

- Correction for multiple substitutions

- ♦ Equal transition/transversion substitution rate

- Equal nucleotide frequencies
  - Constant substitution rate among sites

*Jukes & Cantor (1969)*

$$d_{ij} = -\frac{3}{4} \text{Ln} \left( 1 - \frac{4}{3} \frac{u}{L} \right)$$

- [Gamma model](#) for variable substitution rates among sites

*Jin & Nei (1990)*

$$d_{ij} = \frac{3}{4} a \left[ \left( 1 - \frac{4}{3} \frac{u}{L} \right)^{-1/a} - 1 \right]$$

- Variable nucleotide frequencies  
and constant substitution rate among sites

*Tajima & Nei (1984)*

$$d_{ij} = -b \text{Ln} \left( 1 - \frac{1}{b} \frac{u}{L} \right) \quad b = 1 - \sum_{n=1}^4 g_n^2$$

$g_n$  is the frequency of the  $n$ -th type of nucleotide and is estimated from the data as the average for all the sequences analyzed.

- ♦ Variable transition/transversion substitution rates  
and equal nucleotide frequencies

- Constant substitution rate among sites

*Kimura (1980)*

$$d_{ij} = -\frac{1}{2} \text{Ln} \left( 1 - 2 \frac{u_s}{L} - \frac{u_v}{L} \right) - \frac{1}{4} \text{Ln} \left( 1 - 2 \frac{u_v}{L} \right)$$

where the transition / transversion ration is estimated from the data.

- [Gamma model](#) for variable substitution rates among sites

*Nei (1991)*

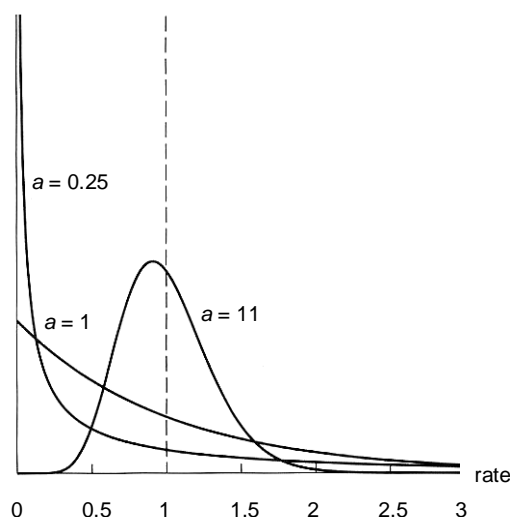
$$d_{ij} = \frac{a}{2} \left[ \left( 1 - 2 \frac{u_s}{L} - \frac{u_v}{L} \right)^{-1/a} + \frac{1}{2} \left( 1 - 2 \frac{u_v}{L} \right)^{-1/a} - \frac{3}{2} \right]$$

Assumptions on evolution model are summarized for each case in the following table:

<i>multiple substitutions</i>	<i>transition/transversion</i>	<i>nucleotide frequencies</i>	<i>rate among sites</i>	
no				simple matching
yes	Equal	Equal	Constant	Jukes & Cantor
			variable	Jin & Nei
		Variable	Constant	Tajima & Nei
		Equal	Constant	Kimura
	Variable		variable	Nei

## Gamma model

Under assumption of equal substitution rate for all sites, a fixed rate, say 1, is assigned to each site. In real data, variations are often observed between sites and this assumption becomes unrealistic. A more suitable model would assign its own rate to each site. Of course we are unable to estimate each of these rates but it would be sufficient to know their distribution.



The gamma distribution is often used as approximate distribution. If the mean is fixed, here to 1, this gamma distribution is specified by a single parameter  $a$  and according to this parameter it can mimic a large variety of distribution shapes. The parameter  $a$  is expressed as the square of the inverse of the coefficient of variation of the substitution rate.

A small value for  $a$  approximates situations where substitutions are rare for most of sites and very frequent only for some ones. A medium value, around 10, simulates situations with nearly symmetrical variations around 1, the variation range decreasing for larger  $a$  values. At the limit, an infinity value for  $a$  comes back to equal substitution rates case.

An accurate estimation of  $a$  would require a substantial number of sequences. It is rarely the case and in DARwin the value for  $a$  has to be assigned by the user.



*For two sequences differing in a great proportion of their sites (75% or more for Jukes & Cantor), the dissimilarity would be infinite. The program*

regards this as an error and stops at the first pair of units causing the problem; a sub-window indicates which pair is concerned and the bootstrap number if error occurs in a bootstrap sample. To avoid this error, it would be necessary to remove too distant units or to limit analysis to more conserved parts of the sequences. If the error occurs in a bootstrap sample, it may sometimes result only from a particular random effect and it may be sufficient to try again with the same data.

## ***Dissimilarity bargraph***

This graphal tool displays the distribution of the dissimilarities for a selected .DIS file. Bargraph parameters can be modified by the user:

- **Start value:** lower limit of the first class.
- **Class number**
- **Step:** class range

Any value lower than the start value is joined to the first class. Any value greater than the superior limit of the last class is joined to this last class.



**Copy** graph to clipboard:

Left click: EMF (vector) format  
Right click: BMP (raster) format



**Print** the graph on the current printer



***This window is multi instantiable and can be opened several times.***

## ***Dissimilarity extreme values***

This function extracts a list of the  $K$  highest or lowest values in a dissimilarity matrix.



**Dissimilarity:** The input file is necessarily a .DIS file. A particular identifier for units can be selected in a companion .DON file.

Options

**Display / Mask Options:**

- **lowest** or **highest** values
- number  $K$  of dissimilarities to display.  $K$  cannot be greater than 5.000. By default the proposed  $K$  is 5.000 or the number of dissimilarities  $(N(N-1)/2)$  if  $N$  is the number of units when this number is lower than 5.000.
- Exclude '1' (or '0'). If checked, only values strictly lower than 1 (or strictly greater than 0) will be listed.

**Text window**

- Click on  $i$ ,  $j$ ,  $Id(i)$ ,  $Id(j)$  or  $d(i,j)$  column header: sort the list in increasing or decreasing order of the column.



**Copy** the text window to the clipboard



***This window is multi instantiable and can be opened several times.***



## Dissimilarity properties

These procedures check that a dissimilarity read in .DIS file in input verifies or not some particular properties (see [method](#)).

### Even dissimilarity (semi-proper)

Verify the relation  $d(i, j) = 0 \Rightarrow d(i, k) = d(j, k)$  for any  $k$ .

The procedure stops at the first triplet which does not verify the relation.

### Distance (Metric)

Verify that the dissimilarity is even and if it is the case verify that for any triplet  $d(i, j) \leq d(i, k) + d(j, k)$

The procedure stops at the first triplet which does not verify the relation.

### Euclidean distance

Verify firstly that the dissimilarity is even and is a distance, and if it is the case, verify that the  $W$  matrix has not negative eigenvalues.

The  $W$  matrix is the scalar product matrix relative to an arbitrary unit  $m$ :

$$W_m(i, j) = \frac{1}{2} (d^2(i, m) + d^2(j, m) - d^2(i, j))$$

This matrix is diagonalized with a simple Jacobi algorithm that returns the eigenvalues.

Optionally, the calculated  $W$  matrix, the diagonalized matrix and the sorted eigenvalue list can be displayed in the text window.



*Eigenvalue approximation is an iterative procedure that stops when a predefined precision is reached, so eigenvalues are only approximated values. Moreover, stored distances are only numbers truncated to a given numerical precision. So it may happen that the smallest eigenvalue appears as negative even if the distance is a real Euclidean distance. However, this negative eigenvalue will be always very small.*

### Additive tree distance

Verify firstly that the dissimilarity is even and is a distance, and if it is the case, verify that for any quartet, among the three sums of distances two by two, the two largest are equal.

The procedure stops at the first quartet which does not verify the relation.

### Ultrametric

verify firstly that the dissimilarity is even and is a distance, and if it is the case, verify that for any triplet, the two greatest distances are equal.

The procedure stops at the first triplet which does not verify the relation.

## Metric index

It can be shown that, if  $d$  is a dissimilarity, it is always possible to find a value  $p$  (between 0 and 1) such that for any  $l$  ( $0 \leq l \leq p$ ),  $d^l$  is a distance. This function returns the  $p$  value that can be used with power [dissimilarity transformation](#) to transform a dissimilarity in a distance.

The  $p$  value cannot be analytically expressed and has to be numerically estimated for each triplet. The algorithm approaches the solution in refining iteratively the extremes of a search interval. The final result will be the smallest value among all triplets, this ensures that the metric condition holds for any triplet.

If only the final result is asked, the superior limit of the search interval for the current triplet is set to the smallest value found on previous triplets. So for a lot of triplets, the metric condition is already verified for this initial value and the procedure stops immediately for this triplet.

**Index distribution.** It may be useful to display the index distribution for all triplets in order to identify some extreme and possibly aberrant triplets. Then the right value has to be estimated for each triplet and the procedure may be longer.

## ***Euclidean index***

It can be shown that if  $d$  is a distance, it is always possible to find a value  $p$  (between 0 and 1) such that for any  $l$  ( $0 \leq l \leq p$ ),  $d^l$  is an Euclidean distance.

In some cases, the  $p$  values are analytically known; for example, it has already been indicated that the square root ( $p = \frac{1}{2}$ ) of a City-Block distance is a Euclidean distance. Often this value is not known and must be numerically estimated on the data.

The algorithm approaches the result in refining iteratively the extremes of a search interval, starting from 0 and the metric index as initial extremes. It has been optimized to limit the needed number of matrix diagonalizations.

This function returns the  $p$  value which can be used with power [dissimilarity transformation](#) to transform a dissimilarity in a Euclidean distance.

A common application involves factorial analysis which requires Euclidean distances. If  $d$  is not Euclidean, a prior transformation is needed. An usual technique is to add a constant to each dissimilarity but this constant often happens to be very large, inducing important distortions. The order-preserving power transformation which less modifies information upon  $d$  seems preferable.

## ***Furnas portraits***

### **Method**

This graphical tool was proposed by Furnas (1989) to illustrate some dissimilarity properties. Let  $d$  be a dissimilarity defined on a triplet  $t$ :  $(a,b,c)$ . This triplet can be located in the three dimensional space corresponding to  $d(a,b)$ ,  $d(a,c)$ ,  $d(b,c)$ . If each dissimilarity is normalized by the sum of the three dissimilarities, then the triplet  $t$  is projected on the canonical plan ( $d(a,b) + d(a,c) + d(b,c) = 1$ ) or more precisely on the triangle intersection between this canonical plan and the 3-dimensions space.

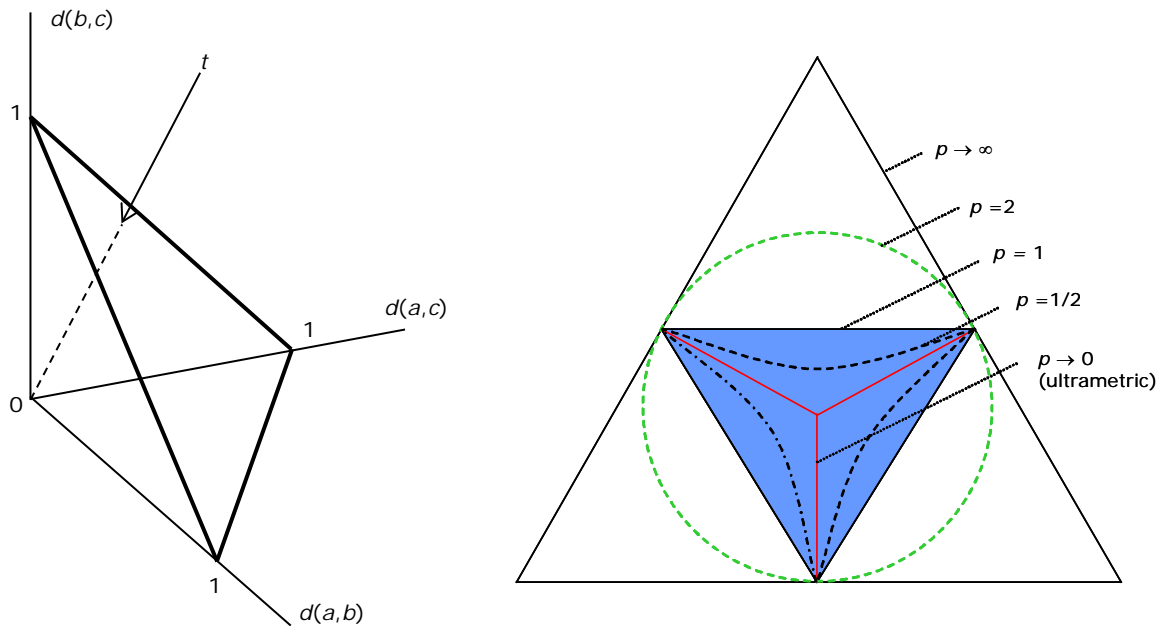
For a set  $E$ , all triplets  $(i,j,k)$  can be positioned on this canonical plan and their distribution in the triangle depends on the conditions satisfied by  $d$  and is related to its metric index.

If the points occupy the whole triangle,  $d$  is a simple dissimilarity without any particular property.

If they are enclosed in the small internal blue triangle,  $d$  is a distance.

In terms of the metric index  $p$ , a simple dissimilarity corresponds to  $p > 1$  and a distance to  $0 < p \leq 1$ . Intermediate  $p$  values can also be mapped on this plan. For instance, if all the points are in the green circle as illustrated on the figure, corresponding to  $p=2$ , then  $d^{1/2}$  is a distance. More generally, the envelope corresponding to a particular  $p$  value can be drawn in the triangle. If all triplets are enclosed in this envelope, then  $d^{1/p}$  is a distance.

At the limit, the envelope corresponding to  $p=0$  is restricted to the red internal 3-branches star that corresponds to the ultrametric.



This graphical tool is useful to detect some aberrant triplets or to approach graphically the metric index.

## Procedure

- Dissimilarity: select a dissimilarity input file



- Partial or full triangle: as each of the three units of a triplet (i,j,k) can be affected either to a, b or c, a same triplet is repeated six times on the graph. So one of the six sub-triangles is sufficient to illustrate all the triplets. However, the full triangle gives a more readable graph but it may be long to draw for large datasets.
- Power: draw on the graph the envelope corresponding to a power numerically defined by the user or adjusted with the cursor. If the animation option is selected, successive user-defined p values are drawn on the same graph.



**Copy** the graph in the clipboard in bitmap format



*This procedure can take long time with more than one hundred units. Partial triangle draws fewer points than full triangle (6 times less) and run faster.*



***This window is multi instantiable and can be opened several times.***

## Dissimilarity transformations

This menu proposes mathematical transformations of an initial dissimilarity d read in input file in a new dissimilarity d' recorded in an output file.

- A first set of transformations is used to translate a similarity in dissimilarity or to create by a monotone transformation a new dissimilarity with some mathematical properties (see [Metric index](#) or [Euclidean index](#)) or to normalize a dissimilarity.

- $D' = a * D + b$  where a and b are user-defined positive parameters
- As dissimilarities cannot be negative, the procedure aborts when a negative d' is found.
- $D' = 1 - D$ : linear transformation from a similarity to a dissimilarity (this transformation cannot be obtained directly with the previous linear transformation that forces a to be positive)
- Power:  $D' = D^p$  where the power p (>0) is user-defined.
- Square root: power transformation with p = 1/2
- Max normalisation:  $D' = D / \text{Max}$  where Max is the greatest observed dissimilarity.



*Multiplication by a positive constant or addition of a constant, as for the first and the fifth transformations, are linear transformations that do not affect the inferred tree structure since tree representations are invariant by linear transformation. It is not the case for power transformations; however these transformations are monotone increasing functions of d. Then d et d'=f(d) are equivalent in order (for any unit quartet,  $d(i,j) \leq d(k,l) \Leftrightarrow d'(i,j) \leq d'(k,l)$ ) and are thus partially of comparable interpretation.*

- A particular transformation adds a **random noise** to a dissimilarity. This function has only a methodological application and can be used to test sensibility of tree methods to data error.

An error level  $e$  is defined by the user as a percent. For each dissimilarity  $d(i, j)$ , a value  $a$  is randomly drawn in a uniform distribution between  $-e$  and  $+e$ . If  $d_m$  is the average dissimilarity between all  $i$  and  $j$ , then

$$d' = d + \frac{a}{100} d_m$$

(a negative  $d'$  is not allowed and in this case, a new  $a$  value is drawn)

- The two last transformations create an **order** dissimilarity from a dissimilarity. These transformations are used only for methodological purposes and have not yet well established properties for biological applications. The underlying assumption is that, as in nonparametric statistical methods, a dissimilarity measure based on orders might be more robust to data errors (Bonnot *et al*, 1996).

**Method** Let  $d$  be a dissimilarity defined on a set  $E$ , let  $i$  and  $j$  be two units of  $E$ . We consider that  $i$  and  $j$  are close if they see other units in the same order.

Let  $u$  and  $v$  be two units of  $E$ , we define  $D_{uv}^o(i, j)$ , a partial order distance between two units  $i$  and  $j$  relatively to  $(u, v)$  according to dissimilarities from  $i$  and  $j$  to  $u$  and  $v$ :

	$d(i, u) < d(i, v)$	$d(i, u) = d(i, v)$	$d(i, u) > d(i, v)$
$d(j, u) < d(j, v)$	0	1/2	1
$d(j, u) = d(j, v)$	1/2	0	1/2
$d(j, u) > d(j, v)$	1	1/2	0

Then the order distance on all pairs  $uv$  is

$$D^o(i, j) = \sum_{u, v} D_{uv}^o(i, j)$$

Equalities and inequalities can be refined in defining a threshold  $\varepsilon$ :  $d$  is equal to  $d'$  if  $d' - \varepsilon \leq d \leq d' + \varepsilon$ ,  $d$  is greater than  $d'$  if  $d > d' + \varepsilon$ ,  $d$  is lower than  $d'$  if  $d < d' - \varepsilon$ .

- **Order distance** a threshold  $\varepsilon$  is user-defined as percent of the dissimilarity average
- **Iterated order distance:** the order distance is a distance and can be transformed again in a new order distance. After some iterations, the distance is no longer modified by order transformation.

## Weighted sum of dissimilarities

This function calculates the sum term by term of several dissimilarity matrices estimated on a same set of units, from several sets of variables. A weight can be affected to each dissimilarity.

If  $d_1$ ,  $d_2$  and  $d_3$  are three dissimilarities on a same set of units and if  $w_1$ ,  $w_2$  and  $w_3$  are the weight factors affected to these dissimilarities, the weighted sum  $d$  is:

$$d(i, j) = \frac{w_1}{w_1 + w_2 + w_3} d_1(i, j) + \frac{w_2}{w_1 + w_2 + w_3} d_2(i, j) + \frac{w_3}{w_1 + w_2 + w_3} d_3(i, j)$$

**Input files: the input files are necessarily .DIS files.**



to add a new dissimilarity file



to remove highlighted dissimilarity file

If all the dissimilarities have not the same number of units, the sum will be calculated only on the subset of units which are present in all input files:

**Unit number:** number of units in the file

**Removed units:** number of units in this file that are not found in other dissimilarities and that have to be removed

**Output file:** a new .DIS file to record the resulting dissimilarity.

**Weight factors and Weights:** a weight factor can be affected to each dissimilarity. It is an integer between 0 and 99999 (by default, all weight factors are 1000). The weight used in the sum for a given dissimilarity is the weight factor of this dissimilarity divided by the sum of all weight factors, so the effective weights are always between 0 and 1 and their sum is always 1.

**Bootstraps:** if the input files include bootstrap matrices, the same weighted sum is calculated on the first bootstrap matrix of each input file, on the second one and so on. If all input files have not the same number of bootstrap matrices, the number of bootstraps in the output file will be the smallest number among input files.

# Factorial analysis

## **Method**

Principal coordinates analysis (PCoA) is a member of the factorial analysis family working on distance matrices. It considers the space of high dimension defined by the distances between units two by two. This space has too high dimension to be readable so PCoA searches for a subspace of low dimension where distances between units are as close as possible to the originate distances. PCoA extracts a first axis (one dimension) such that  $\sum_{i,j} (d_{ij} - \delta_{ij})^2$  is minimum (where  $d_{ij}$  is the observed distance between  $i$  and  $j$ ,  $\delta_{ij}$  is the distance between the projections of  $i$  and  $j$  on this axis). Then it extracts a second axis, orthogonal to the previous one (independence condition) minimizing the squared differences and so on. Solutions are given by eigenvectors and eigenvalues of the matrix  $W$  of scalar products between elements that is defined from the  $d_{ij}$  according to the Torgerson formula:  $W_{ij} = - (d_{ij}^2 - d_i^2 - d_j^2 + d^2)/2$ .

The output is the list of coordinates of each unit on each axis. In general, the first axes (3 or 4) summarize a large part of the complete space information and plans of axis 1-2, 1-3, 2-3... are sufficient to exhibit the main structure of the data. The part of information retains by each axis is given by the percent inertia (the eigenvalue of this axis on the sum of all eigenvalues).

The coordinate of a unit on an axis is given by the projection of this unit on this axis. But this unit is well represented by its projection only if the distance between the unit in the full space and its projection on the axis is small. The quality of representation is given by the squared cosinus of the angle between the unit and its projection (small distance  $\rightarrow$  small angle  $\rightarrow \cos^2$  close to 1, large angle  $\rightarrow \cos^2$  close to 0). These  $\cos^2$  are used to avoid misinterpretation of unit neighbourhood on the plans.

PCoA is related to multidimensional scaling methods (MDS) which search for a unique decomposition on a fixed number of axes rather than decompositions axis by axis successively. For a given number of axes, MDS may produce slightly better results than PCoA but these iterative methods require considerable time to treat large numbers of units so only PCoA was developed in DARwin.



*Factorial analysis and tree methods constitute two very different approaches for the representation of diversity structure. Factorial methods aim mainly to give an overall representation of diversity and are not really interested in individual effects. On the other hand, tree methods tend to represent individual relations faithfully and may be less accurate for the overall structure. They are thus two different ways of viewing the data and must be considered complementary rather than concurrent.*



PCoA requires an Euclidean distance, if this property is not verified ([Dissimilarity properties](#)), the [Euclidean index](#) gives the power transformation required to obtain an Euclidean distance ([Dissimilarity transformations](#)).

## Analysis

### Input / output files

The input file is necessarily a .DIS file.

In output, two files are created:

- **.AFT file**

This file stores the coordinates of each unit on the retained axes. It has exactly the same structure as [.VAR file](#) except the signature in the first line: @DARwin 5.0 – AFT.

Eigenvalue and inertia% for each axis are recorded in the comment field.

- **.DON file for  $\cos^2$  [optional]**

If the option is checked, the values of  $\cos^2$  are stored exactly as external identifiers in a new [.DON file](#) with a column for each retained axis. This file can be used in the graphical display to add  $\cos^2$  values to the current identifier in order to facilitate graph interpretation.

Its name will be automatically generated as the .AFT file name followed by \_COS and with extension .DON.

### Parameters

- Number of axes to edit: number of first axes retained in the outputs

### Unit selection



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

### Text window

In output, a text window summarizes the analysis results:

- file name and list of selected units
- a warning if the distance is not Euclidean
- eigenvalue and inertia% for each retained axis
- list of coordinates and  $\cos^2$  for each unit on each retained axis.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.



## Graphical window

Show the resulting PCoA if 'Display when done' is checked.

## Graphical display

- This window is automatically opened at the end of the analysis when the option 'Display when done' is selected.
- When directly call from the main menu, this function asks for an .AFT file name.

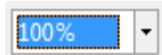
## Graph parameters



**Open** a new .AFT file



**Select** axes in **X** or in **Y** (by default, the graph displays the plan of axes 1 and 2)



**Zoom:** select zoom factor in list box from 100% to 1 000%

## Identification and illustration

By default, units are identified by their numerical identifier in the .AFT file.



**Toggle units labels** display



**External identifiers**

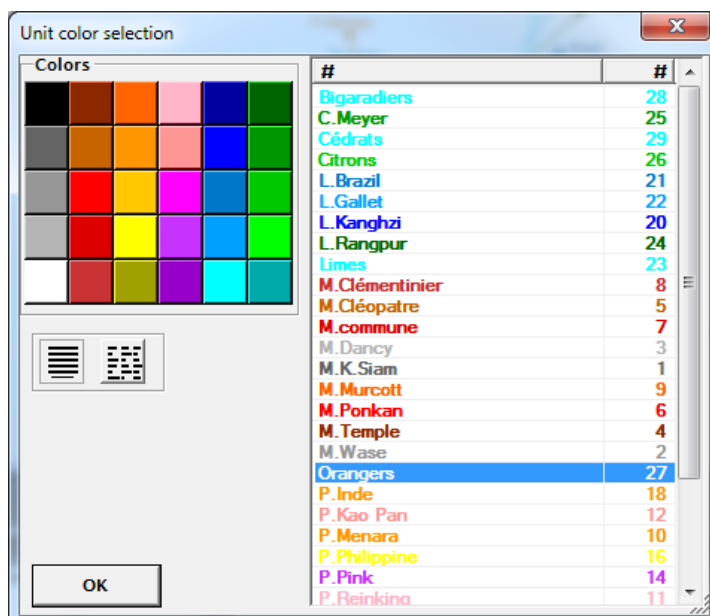
'Open identifiers file' to select a .DON file  
select an identifier in the proposed list



**cos<sup>2</sup>:** (available only if the corresponding xxx\_COS.DON file is found)  
Add to the current identifier (unit number or external identifier) the cos<sup>2</sup> value for a selected axis,  
ex: A10(C<sup>2</sup>1:649) : cos<sup>2</sup> = 0.649 on axis 1 for unit A10  
Single click: choice of the axis cos<sup>2</sup> to display



Open the "**Unit color selection**" window




The list of units is identified by the current identifier (unit number or external identifier), this list can be sorted by identifier (left column) or by unit number (right column), ascendant or descendant by clicking on column header


Selection of a subset of units (shift or control for multiple selection)


Moving the cursor on a color applies temporary this color to the selected units.

Click on the color button to apply the color to the selected units.

This window is sizeable

 Display units with identifier and number (default mode) and column widths are sizeable

 Display only the unit identifier on multi

 Apply and record color selection. The 'close window' button will discard any changes



**Toggle axis graduations** display



**Title, sub-title, comment**

Click on "Show / Hide" toggle title display

Click on other item (Title, Sub title, Comment) to create, modify or delete the texts



Open Windows selection **Font** and **size** choice for texts in the graph



Display the comment field of the .AFT file in input

## Graph exportation



Record the graph in a **.EMF file** (Windows enhanced metafile)



**Copy** bitmap graph in the **clipboard**



Open **print** window for selecting printer and options to print the graph



***This window is multi instantiable and can be opened several times.***

---

# Tree construction

## *Method*

The principle of any tree construction method is to approach as closely as possible the dissimilarity  $d$ , chosen for its relevance in describing the relationships between units, by a distance  $\delta$  that can be represented as a tree.

To find the exact solution, we must enumerate all possible tree topologies for  $n$  units. For each topology, a fit criterion is calculated from edge length estimations and finally, we will retain the tree topology which optimizes this fit criterion. The number of different binary tree topologies over  $n$  units is  $\prod_3^n (2i-5)$ . For  $n = 10$ , there are more than  $2 \cdot 10^6$  different trees and more than  $2 \cdot 10^{20}$  for  $n = 20$ . It is thus impossible to enumerate all the trees when  $n$  goes beyond some tens, even with the most powerful computers. In such situations, the only possibility is to construct, from reasonable heuristics, solutions that will be the best possible but that can never be guaranteed to be optimal.

DARwin proposes methods based on the common agglomerative heuristic that proceeds by successive ascending agglomerations. Initially, the matrix treated has as many elements as units in the population and the tree has a star structure. At each iteration, two elements, units or groups already formed, are defined as neighbours and joined to form a node in the tree. This node is a new fictive element which replaces the two combined elements. The two edge lengths between this node and the two combined elements are estimated. The matrix is updated by estimation of a dissimilarity between this new element and the other elements and its dimension is reduced by one unit. The process is reiterated until all the units are in a single group. The proposed methods are characterized by different choices at the three key points of each iteration: 'neighbourhood' definition, dissimilarity matrix updating, and edge length estimations.

### **'Neighbourhood' definition**

A natural definition of neighbourhood is to declare elements  $i$  and  $j$  as neighbours if  $d(i,j)$  is the smallest dissimilarity. This definition is used in hierarchical clustering like UPGMA. Saitou and Nei (1987) proposed a criterion of neighbourhood based on a principle of parsimony. They consider that edge lengths represent mutational event numbers and, according to the parsimony principle, the total number of mutations, and thus the total length of the tree, must be minimized. They derive a definition of relative neighbourhood,

defined by minimizing a criterion  $Q(i,j)$ , function of  $d(i,j)$  and of the average of dissimilarities from  $i$  and  $j$  at the  $n - 2$  other elements  $k$ :

$$Q(i, j) = d(i, j) - \left( \sum_k [d(i, k) + d(j, k)] \right) / (n - 2)$$

This criterion has properties of optimality in terms of least squares. It can be interpreted, very generally, as a weighting of the dissimilarity between two elements by their dissimilarities to other elements. In term of similarity, this means that two related units which differ largely from other units are more similar than two related units which are equally related to other units. This attitude is justified in many cases and explains the success of the method.

Sattath and Tversky (1977) adopt a very different approach. They start from the characterization of an additive tree distance by the four point's condition. For a quartet  $(i, j, k, l)$ , if  $\delta(i, j) + \delta(k, l)$  is the smallest of the three sums of distances two by two, then the two largest are equal:  $\delta(i, k) + \delta(j, l) = \delta(i, l) + \delta(j, k)$ . The initial dissimilarity  $d$  is not an additive tree distance but it is always possible to form the sums of dissimilarities two by two. Among these three sums, one is the smallest,  $d(i, j) + d(k, l)$ , for example; the pairs  $(i, j)$  and  $(k, l)$  are thus considered good candidates to be neighbours. They are assigned a score of 1, while other pairs  $(i, k)$ ,  $(j, l)$ ,  $(i, l)$  and  $(j, k)$ , are attributed a null score. All the quartets are scanned and the elements of the pair of largest total score are defined as neighbours. This definition of topological neighbourhood is ordinal in nature, since it depends only on the order of the three sums, and not directly on the dissimilarity values.

## Dissimilarity updating

Given  $i$  and  $j$  the elements (units or groups of units) combined in a new element  $s$ ,  $c_i$  and  $c_j$  the unit numbers of these elements, and  $k$  another element, a general definition of  $d(s, k)$  is the weighted average of dissimilarities between  $k$  and elements  $i$  and  $j$ :

$$d(s, k) = \lambda_i d(i, k) + \lambda_j d(j, k) \quad \text{with} \quad \lambda_i + \lambda_j = 1$$

Elements  $i$  and  $j$  are groups of  $c_i$  and  $c_j$  units and to the weights  $\lambda_i$  and  $\lambda_j$  correspond weights  $w_i$  and  $w_j$  attributed to each unit of elements  $i$  and  $j$ , then:

$$d(i, k) = \frac{\sum_{u \in i} w_i d(u, k)}{\sum_{u \in i} w_i} = \frac{1}{c_i} \sum_{u \in i} d(u, k) \quad , \quad d(j, k) = \frac{1}{c_j} \sum_{u \in j} d(u, k)$$

and

$$\lambda_i = \frac{c_i w_i}{c_i w_i + c_j w_j} \quad , \quad \lambda_j = \frac{c_j w_j}{c_i w_i + c_j w_j}$$

**'unweighted solution'**: a first solution is to define weights  $\lambda$  as function of the element numbers  $c_i$  and  $c_j$ :

$$d(s, k) = \frac{c_i}{c_i + c_j} d(i, k) + \frac{c_j}{c_i + c_j} d(j, k)$$

Despite its mathematical formulation where a weight is affected to elements, it corresponds to a criterion said **'unweighted'** that refers to a same unitary weight given to all units belonging to  $i$  and  $j$  since the formula is obtained for  $w_i = w_j = 1$ .

**'weighted solution'**: a second solution is to define a simple arithmetic average between the elements themselves:

$$d(s, k) = \frac{1}{2} [d(i, k) + d(j, k)]$$

which corresponds to a **'weighted'** criterion since it is obtained if elements of  $i$  and  $j$  receive different weights:  $w_i = 1/c_i$  and  $w_j = 1/c_j$

The choice between these two criteria depends on the nature of the population studied. If the whole arises from a real process of sampling on a given structure of diversity, then the number of units in each element has a meaning and must be taken into account in the dissimilarity estimations. On the other hand, if the population does not result of a real sampling procedure, the size of each group has no real meaning and has not to be taken into account.

Some other criteria are sometimes used for hierarchical clustering:

. single linkage:  $d(s, k) = \min[d(i, k), d(j, k)]$

. complete linkage:  $d(s, k) = \max[d(i, k), d(j, k)]$

Except in very particular cases, it is difficult to justify these criteria.

. Ward criterion: it concerns only Euclidean distances and is based on geometric considerations. It can be seen as a method that searches at each step a local optimum to minimize the within-group or equivalently to maximize the between-group inertia.

The distance between two elements is the weighted square of the Euclidean distance between their gravity centres:

$$\delta(i, j) = \frac{c_i c_j}{c_i + c_j} d^2(g_i, g_j)$$

where  $g_i$  and  $g_j$  are the gravity centres of groups  $i$  and  $j$ .

Then:

$$\delta(s, k) = \frac{(c_i + c_j)\delta(i, k) + (c_j + c_k)\delta(j, k) - c_k\delta(i, j)}{c_i + c_j + c_k}$$

## Edge lengths

Let  $l(i,s)$  and  $l(j,s)$  be the edge lengths between elements  $i$  and  $j$  combined to form the node  $s$ . For adjustment to an ultrametric,  $d(i,j)$  is simply divided equally between the two edges:

$$l(i,s) = l(j,s) = d(i,j)/2$$

For adjustment to an additive tree distance,  $d(i,j)$  is divided proportionally to the mean dissimilarities of  $i$  and  $j$  to all other elements  $k$ . This can be written:

$$l(i,s) = (d(i,j) + e)/2 \text{ and } l(j,s) = (d(i,j) - e)/2 \text{ where } e = (\sum_k [d(i,k) - d(j,k)])/(n-2)$$

DARwin proposes different versions of the classical methods: hierarchical clustering, Neighbor-Joining, Score. In complement it proposes also some new algorithms. Neighbor-Joining under topological constraints is a modification of Neighbor-Joining where some unit subsets may have a predefined topology. Ordinal Neighbor-Joining and ordinal Score are modified versions based on an ordinal transformation of the dissimilarities. 'Influent unit detection' is a form of jackknife on Neighbor-Joining method.

## Bootstraps

If the dissimilarity file includes [bootstrapped matrices](#), tree construction methods use the trees inferred from these bootstrapped dissimilarities to assess the uncertainty of the tree structure. Concretely, a bootstrap value is given to each edge that indicates the occurrence frequency of this edge in the bootstrapped trees.

Two approaches can be used to summarize bootstrapped trees. A first one produces a majority-rule consensus of the bootstrapped trees. According to the majority rule, only edges which are found in more than 50% of the trees are present in the consensus. So a bootstrap value (in percent) varies between 50 and 100. A second approach, retained in DARwin, is to consider that the best tree is not the consensus tree but, indeed, the tree inferred from the initial data. So the produced result is the initial tree where each edge receives a bootstrap value corresponding to its occurrence frequency in the bootstrapped trees. Then bootstrap values range between 0 and 100. Moreover, a dissimilarity between each bootstrapped trees and the initial tree is estimated as the fraction of edges that are present in one tree and not in the other one (the 'edge distance' equivalent to the Robinson and Foulds distance). The mean and the 95 and 5 percentiles of these dissimilarities are calculated, to describe the dispersion of the bootstrapped trees around the initial tree.

# ***Hierarchical tree***

## **Method**

The family of methods often described as hierarchical clustering corresponds to the definition of neighbourhood according to the minimal dissimilarity, with an adjustment to an ultrametric, and various formulae are proposed for updating. Among these, average or weighted average are the most commonly used and the corresponding methods are referred to as UPGMA and WPGMA, for unweighted or weighted pair group method using average. Criteria of simple linkage, complete linkage and Ward are also offered.

At each iteration, the pair of elements with the smallest dissimilarity is grouped to form a new node. If this minimal value is shared by several pairs, they are grouped at the same iteration. If several retained pairs involve identical elements, the new node groups all the involved elements. For example, if (12,20) and (15,20) are retained, the node will group 12,15 and 20. So multifurcations may be created in the tree and some nodes may have a degree greater than 3. To take into account some imprecision on dissimilarities, the strict equality between minimal distances can be extended to an interval  $[d, d + \varepsilon d]$ : if  $d$  is the smallest dissimilarity observed for this iteration, all pairs with  $d' \leq d + \varepsilon d$  will be grouped at this step.

A supplementary node of degree 2 is added for the root of the tree. Normally, a node corresponds to a divergence and its degree is at least 3. This particular node is only used to allow representing the tree as a dendrogram.

## **Procedure**

### **Input / output files**

- The input file is necessarily a .DIS file.
- In output, a .ARB file stores the structure and the parameters of the tree according to [.ARB file structure](#).

### **Parameters**

- [Single linkage, Complete linkage, Ward, UPGMA or WPGMA](#) to select the method
- [Threshold equality %](#) expressed in percent of the minimal dissimilarity for a given iteration

### **Unit selection**



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate the selection.

## Text window

A text window displays the main results:

- the list of selected units
- for each iteration: the number of remaining elements, the minimal dissimilarity for grouping, the new node number and its connected elements
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Neighbor-Joining

### Method

The Neighbor-Joining method proposed by Saitou and Nei (1987) is often used in genetic diversity. It uses the criterion of relative neighbourhood, **weighted** average for dissimilarity updating, and adjustment to an additive tree distance. The algorithm complexity is in  $n^3$  and allows to analyse matrices of some hundreds of units. Gascuel (1997) proposes UNJ (**unweighted** Neighbor-joining), which uses a criterion of weighted average. As the neighbourhood criterion calculated for each pair results of a complex formula involving all dissimilarities, the probability of equal criterion for several pairs is very low. So the implemented algorithm groups only one pair at each iteration and the trees are always binary trees.

**MVR** (maximum variance reduction), proposed by Gascuel (2000), is a NJ method where formula for dissimilarity updating is modified. It considers that observed dissimilarities are only estimates of the unknown true dissimilarities. Then in the general updating formula:

$$d(s,k) = \lambda_i d(i,k) + \lambda_j d(j,k)$$

the precision on dissimilarities could be taken into account in increasing the weight for well estimated dissimilarities and conversely in decreasing the weight for poor estimations.

MVR retains an error model where an observed  $d(i,j)$  is the sum of the true value  $D(i,j)$  and an error  $\varepsilon(i,j)$ , errors are assumed independent and normally distributed with a variance  $V(d(i,j))$ . These variances are used to estimate the weights  $\lambda$ .

This method finds an application for sequence data when dissimilarities with correction for multiple substitutions are used since variances of corrected dissimilarities are known (see [Dissimilarity for Sequence data](#)).



## Procedure

### Input / output files

- The input file is necessarily a .DIS file.
- In output, a .ARB file stores the structure and the parameters of the tree according to [.ARB file structure](#) including bootstrap values.

### Parameters

- [Weighted, Unweighted](#) or [MVR](#)  
If MVR is selected, a file with the same name as dissimilarity but with .VRD extension is automatically opened

### Bootstraps

When the .DIS file includes bootstrapped dissimilarities, this option offers to estimate edge [bootstrap values](#). (bootstrap analysis is not available for MVR method)

### Unit selection



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate the selection.

### Text window

A text window displays the main results:

- the list of selected units
- for each iteration: the two elements grouped at this step and the corresponding node
- the list of edges and their length
- in case of bootstrap: the list of bootstrap values for each internal edge, the average of [‘edge’ distances](#) between bootstrapped trees and the initial tree, the 5 and 95 percentiles of these distances.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.



### **New in DARwin 6 – Bootstrap computing**

As bootstrapped trees are independent, they can be distributed between logical cores to improve computing time.

A textbox informs on computing steps and a ‘Cancel’ button is available to abort the procedure. A main progress bar indicates global progression and secondary progress bars indicate the progression on the different cores (only for the 4 first cores)

# Scores

## Method

Sattath and Tversky (1977) propose the Score algorithm which uses a topological neighbourhood definition, weighted average for dissimilarity updating, and an adjustment to an additive tree distance. The complexity in  $n^5$  is higher than that of Neighbor-Joining. This method may be very long for large data sets but it could be preferred for its topological point of view.

At each iteration, all quartets are analysed to fill the matrix of scores between pairs.

The two pairs involved in the smallest sum among the three sums of dissimilarities two by two receive a score of 1 and the four other ones, a score of 0. If the two smallest sums are equal, the four involved pairs receive a score of 1/2. If the three sums are equal, each pair receives a score of 1/3.

The pair of maximal score is grouped to form a new node. If several pairs reach this maximal score, they are grouped at the same iteration. If several pairs of maximal score involve identical units, the new node groups all the involved units. For example, if (12,20) and (15,20) have a maximal score, the node will group 12,15 and 20. So multifurcations may be created in the tree and some nodes may have a degree greater than 3.

As the complexity is in  $n^5$ , bootstrap analysis would be impracticable even for moderate size data, it was not implemented in DARwin.

## Procedure

### Input / output files

- The input file is necessarily a .DIS file.
- In output, a .ARB file stores the structure and the parameters of the tree according to [.ARB file structure](#).

### Unit selection



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

### Text window

A text window displays the main results:

- the list of selected units
- for each iteration: the matrix of scores between pairs (only if data have less than 30 units), theoretical maximal score for the number of

- remaining elements, maximal score really found, the pairs of elements corresponding to this maximum and their nodes
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## ***Ordinal Neighbor-Joining***

### **Method**

The underlying assumption is that, as in nonparametric statistical methods, a dissimilarity measure based on orders might be more robust to data errors (Bonnot *et al*, 1996). The ordinal point of view is here to consider that two elements  $i$  and  $j$  are close if they see other elements in the same order. An ordinal dissimilarity  $d^o$  between two elements  $i$  and  $j$  is defined from orders of distances from  $i$  and  $j$  to the others elements (see [Dissimilarity transformations](#)).

The ordinal Neighbor-Joining method is not a classical Neighbor-Joining on an order dissimilarity after preliminary order transformation of the initial dissimilarity. Indeed, the dissimilarity updating at the end of each iteration would not have real meaning on order dissimilarities and the new dissimilarities have to be estimated from the non transformed values. This reduced matrix will be order-transformed at the beginning of the following step. So the order transformation takes place only in the neighbourhood definition, two elements  $i$  and  $j$  are regarded as neighbours if they see the other elements in a same particular way.



*Experimentally on simulated data, we showed that the ordinal Neighbor-Joining method has to be used only when a high level of random noise is suspected. In other cases, it may induce some loss of efficacy.*

### **Procedure**

#### **Input / output files**

- The input file is necessarily a .DIS file
- In output, a .ARB file stores the structure and the parameters of the tree according to [.ARB file structure](#) including bootstrap values.

#### **Parameters**

[Weighted, Unweighted](#)

#### **Unit selection**



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection

### Text window

A text window displays the main results:

- the list of selected units
- for each iteration: the two elements grouped at this step and the corresponding node
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Ordinal Scores

### Method

The approach is the same as for [Ordinal Neighbor-Joining](#).



*As the definition of neighbourhood for Score method is yet ordinal in nature, the ordinal version has not shown real interest for data analysis. It is proposed here only for methodological purposes.*

### Procedure

#### Input / output files

- The input file is necessarily a .DIS file.
- In output, a .ARB file stores the structure and the parameters of the tree according to [.ARB file structure](#).

#### Unit selection



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

### Text window

A text window summarizes the analysis results:

- the list of selected units
- for each iteration: the matrix of scores between pairs (only if data have less than 30 units), theoretical maximal score for the number of remaining elements, maximal score really found, the pairs of elements corresponding to this maximum and their nodes
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

# ***Neighbor-Joining under topological constraints***

## **Method**

This modified version of Neighbor-Joining (the weighted version of Saitou and Nei) allows forcing the a priori known topology of one or several unit subsets such that the topologies of these subsets will be respected in the tree inferred on the whole set. The topology of a subset is described by a tree structure which is read in a .ARB file and is translated in a list of bipartitions which will be considered as constraints by the algorithm. If several unit subset constraints are defined and if the intersection between these subsets is not empty, the constraints on these intersection units must be compatible.

As for classical Neighbor-Joining, the pair optimising the criterion defines the two neighbours. Grouping these two elements forms a bipartition: units belonging to these two elements against all other units. If this bipartition is compatible with the bipartitions defined as constraints, the group is accepted. In contrary case, this group is refused and the following pair optimising the criterion is examined until a compatible pair is found. If no more pair satisfies the constraints, all remaining elements are grouped to the same internal node, giving a local star structure.

## **Applications**

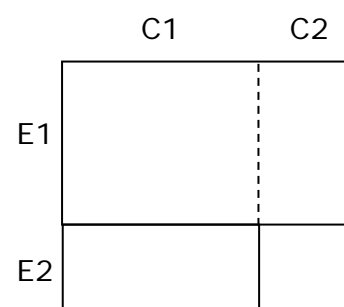
This method finds a large range of applications.

### **- Structured populations**

For example, let H be a collection of genotypes where several subsets are known as being geographically isolated for a long time. Each of these subsets has followed its own independent evolution process. Then it appears illogical that the tree structure inferred for a given subset might be under influence of other subsets as it is the case with NJ where neighbourhood definition implies all units. A solution would be to apply NJ separately to each of these subsets and to specify these trees as constraints in the analysis of the whole dataset, in order to describe the relations between these subsets and with other possible units not implied in the subsets.

### **- Structural missing data**

Let E be a unit set where a first part E1 is characterized by a set of variables and a second part E2 cannot be observed for a part C2 of the variables (for example a large deletion in common for the whole subset). A common analysis of the two subsets would concern only the common variables C1



but all information coming from C2 would be lost. Another solution is to construct a tree on E1 with dissimilarities calculated on C1 + C2 variables. This tree inferred from all the variables is assumed to be a better solution for this subset. Then this tree is specified as constraint for an analysis on E1+E2 based on dissimilarities calculated only on C1.

#### - Multiple outgroups

Outgroups by definition are relatively distant units and they often disturb the resulting tree. A solution would be to graft a posteriori these outgroups on a tree build on the active units (see [Grafting](#) function). When several outgroups are used, we may be also interested by their own structure that will not be revealed by grafting one by one. A better solution could be to apply a tree construction method only to the active units. This inferred tree is not disturbed by the outgroups and is expected as it in the tree including also the outgroups. For that, it will be specified as constraint in an analysis including the active units and the outgroups.

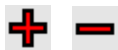
## Procedure

### Input / output files

- The input file is necessarily a .DIS file.
- In output, a .ARB file stores the structure and the parameters of the tree according to .ARB file structure.

### Parameters

- Constraint tree selection:



to add or remove trees of constraints as .ARB files

The procedure aborts if the constraints describes by these trees are not compatible.

### Unit selection



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

### Text window

A text window displays the main results:

- the list of selected units
- the list of constraints expressed as bipartitions: a unit subset against its complementary subset
- for each iteration: if it is the case, the list of pairs which violate a constraint and are not retained, the first pair that satisfies the constraints and the corresponding node
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## ***Influential unit detection***

### **Method**

A unit is judged as influential on a tree structure if the tree build after removal of this unit is different in topology from the initial tree. In a jackknife-like procedure, each unit is successively removed and a partial tree is inferred. Then the removed unit is grafted on this partial tree (see Grafting procedure) to restore a tree on the same unit set. This tree and the initial tree are compared and the quartet distance (see Tree comparison – Quartet distance) between these trees is regarded as a measure of influence.

If the quartet distance is high, the trees inferred with or without this unit are very different and the unit is judged as influent. Conversely, if the distance is small, the two trees are similar and the unit has limited influence on the tree structure.

The interpretation of this influence measure is somewhat tricky. A very influent unit may be an important unit which contributes to structure the unit set. It may be also a very particular unit which is too different from the others and would logically be removed from the data. Inversely, a unit with only a small influence may be an intermediate unit without particular characteristics, the tree structure being imposed by other more influent units. But it may also belong to a group of very influent units, the structure being maintained by these other units even if this unit is removed.

This algorithm proposes two methods:

- weighted Neighbor-Joining associated to a mean square grafting procedure
- Score tree method associated to a score grafting procedure.

### **Procedure**

#### **Input file**

- The input file is necessarily a .DIS file.

#### **Parameters**

- Method:
  - Weighted Neighbor-Joining and mean least square grafting
  - Score tree and score grafting

#### **Unit selection**



Buttons to select / unselect part or all units

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

## Text window

- tree construction for the whole unit set, as for NJ or Score
- numbers of resolved and unresolved quartets for this tree
- a table giving for each removed unit (sorted in increasing order of the quartet distance):
  - R resolved quartets for the grafted partial tree
  - U unresolved quartets for the grafted partial tree
  - R/Rs resolved quartets with the same topology
  - R/Ru resolved quartets with different topologies
  - R/U resolved quartets in a tree and unresolved in the other one
  - U/U quartets unresolved in the two trees
  - Qd the quartet distance between the two trees
- the list of edges and their length



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.



## New in DARwin 6 – Computing

Removing of successive units are distributed between logical cores to improve computing time.

A textbox informs on computing steps and progression is illustrated by a progress bar. A 'cancel' button is available to abort the procedure.



*The quartet tree distance estimation procedure takes long time for big datasets. As this procedure is called for each unit of the dataset to compare initial tree and tree obtained without this unit then grafted, complete exploration of the whole dataset can take several hours with more than 300 units.*

---



# Trees...

## **Draw**

Display a tree read in a .ARB file in input. Several functions are proposed to modify the tree representation and facilitate its interpretation (color, label, rotation, root...). All the user choices are recorded in the tree file (see [.ARB file components](#)) that allows to redisplay the tree exactly as it was saved.

Two families of functions are proposed:

- **actions**

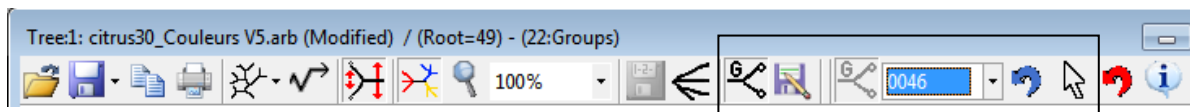
They are applied at the whole tree and do not depend on the choice of a particular element of the tree: mirror, external identifiers, police, titles...

- **tools**

They are applied to particular nodes and require a user action to select these nodes (root, zoom, rotation, branch swapping...).

The selection of a tool gives to the cursor a particular shape. Then a node can be selected directly on the graph in clicking with the cursor on this node or chosen in a list.

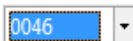
A special zone is opened at the right of the toolbar, for example for the edge contraction tool:



where



Icon identifying the current tool



Open a node list for selecting the node by its number



Undo all previous actions of this tool



Unselect the current tool

Two particular functions are also included in the toolbar because they are applied interactively on the tree itself and work like tools. They do not modify the tree representation but create new information:

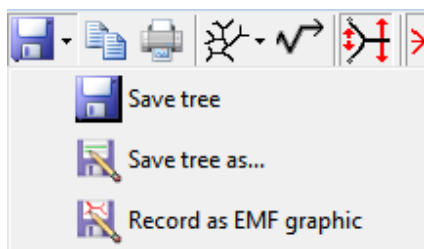
- . Edge contraction modifies the structure of the tree giving a new tree which is recorded in a new tree file.

- . Group definition records a group identifier in a .DON file.

## **Tree draw toolbar**



**Open** a new .ARB file



### Record tree and graph

- **Save tree:** save the tree (and the edition actions, see [.ARB file components](#)) in the same .ARB file
- **Save tree as...:** save the tree (and the edition actions) in a new .ARB file
- **Record EMF graph:** record the tree graph in a .EMF file (Windows enhanced metafile)



### Copy bitmap graph to clipboard

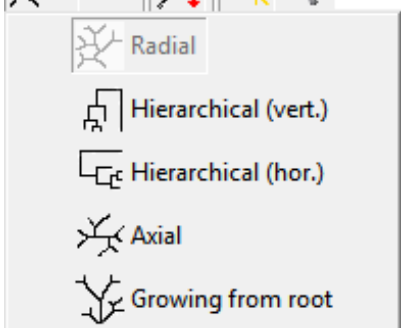
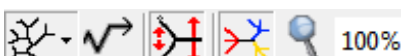
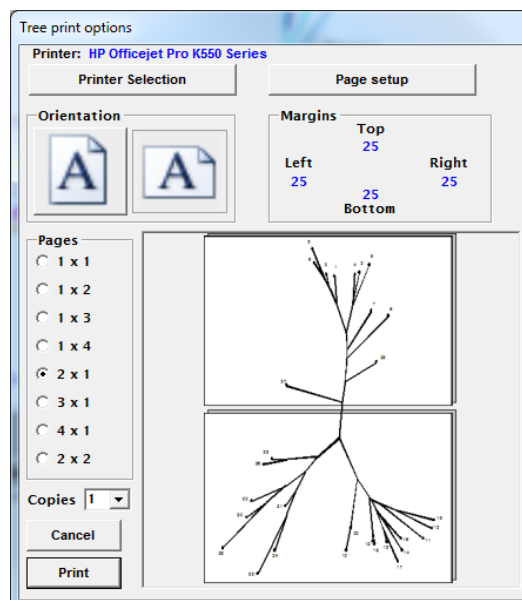


### Print the graph on the current printer

For trees with large number of units, it is often useful to print the graph on several pages. DARwin proposes until 4 pages for printing:

- successive horizontal pages: 1x2, 1x3, 1x4
- successive vertical pages: 2x1, 3x1, 4x1
- a square of 2x2 pages.

A small margin is left between pages for assembling.




### Tree representation

Select the mode of representation.

As long as a new root has not been selected (see below), the root by default is:

- the node of degree 2 if it exists (as it is the case for hierarchical clustering),
- the last node remaining after successive concentric pruning of external edges, this choice allows to fill at best the window space for radial representation.




**Root selection:** toggle tool  to change the current root (the node in red) by clicking on another node or by selecting its number in the node list.





**Edition:** toggle the tree edition toolbar witch appears vertically on the upper left side of the tree window.




**Rotation:** tool  (only for radial representation)


- on the root to rotate the whole tree
- on a node to rotate the edge ending at this node
- left button: left (anti-clockwise) rotation 1°
- shift + left button: left (anti-clockwise) rotation 5°
- right button: right (clockwise) rotation 1°
- shift + right button: (clockwise) right rotation 5°


 **Branch swapping:** toggle tool  for branch swapping by clicking on the node or choosing its number in the node list.


 **Horizontal mirror**


 **Vertical mirror**


 **Identification and illustration:** toggle the toolbar witch appears on the left side of the tree window under the edition toolbar.


 **Unit identifiers:** toggle for displaying or hiding unit identifiers


 **Nodes:** toggle for displaying or hiding internal node numbers


 **Topology:** toggle for setting all edge lengths to one in order to exhibit the tree structure in case of very short edges


 **Bootstraps:** is active only in case of bootstrapped trees  
Open a list to choose a minimal threshold for bootstrap displaying. The first item is "No" (no bootstrap display), the second item is "All".


 **Reticulations:** is active only when the tree file includes [reticulations](#)  
Open a list to choose the number of reticulations displayed on the tree graph, the list gives for each reticulation the two linked nodes and the percent of decrease of the LS criterion, the list is sorted on decreasing values of this criterion. The first item is "No" (no reticulation display), the second item is "All".

 **Scale:** toggle for displaying or hiding scale for edge lengths

 **Unit colours:** open a window for choosing particular colours for selected subsets of units: [Unit color selection](#)  
- selection of a subset of units identified by their current identifier (numerical or external identifier) with usual shift and control function for block or unitary selection  
- choice of a colour for this subset (select black to undo a colour)

 **External identifiers**  
Click on "Open identifiers file" to select a .DON file  
Each identifier is added to list items  
Click on identifier item: select identifier as unit label

 **Title, sub-title, comment**  
Click on "Show / Hide" toggle title display  
Click on other item (Title, Sub title, Comment) opens edit window to create, modify or delete the texts

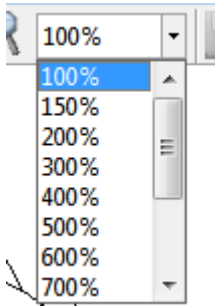
 **Font:** font, style, size for texts (unit identifiers, node numbers, titles...)

 **node zoom:** toggle tool  
Logical zoom that shows in full window the sub-tree below (relatively to the

root) a selected node.



Left or right button on a node for zoom in or out or selection in the node list.



**zoom:** Physical zoom that magnifies a part of the graph window  
- select zoom factor in list box from 100% to 2000%



**Undo:** restore initial tree draw without any identification, illustration or edition actions.

## Internal edge contraction

This function creates a new tree (in a new .ARB file) where some internal edges are contracted (external edges cannot be contracted). The resulting tree shows multifurcations, and at the limit, if all internal edges are contracted the tree is a star.

This function is used to remove internal edges regarded as unreliable because they have too short length or are not strongly supported by bootstrap analysis. It is preferred to retain a multifurcation that shows an uncertainty rather than to maintain an edge that may be inaccurate.



button in the toolbar: toggle tool



- Left clicking on a node (or selecting it in the node list) contracts all edges of the sub-tree rooted on this node relatively to the current root of the tree. The contracted edges appear as dashed lines.

- Left Clicking on a node in a sub-tree previously contracted undoes the contraction for the part of the sub-tree rooted on this node and for the edge ending at this node.

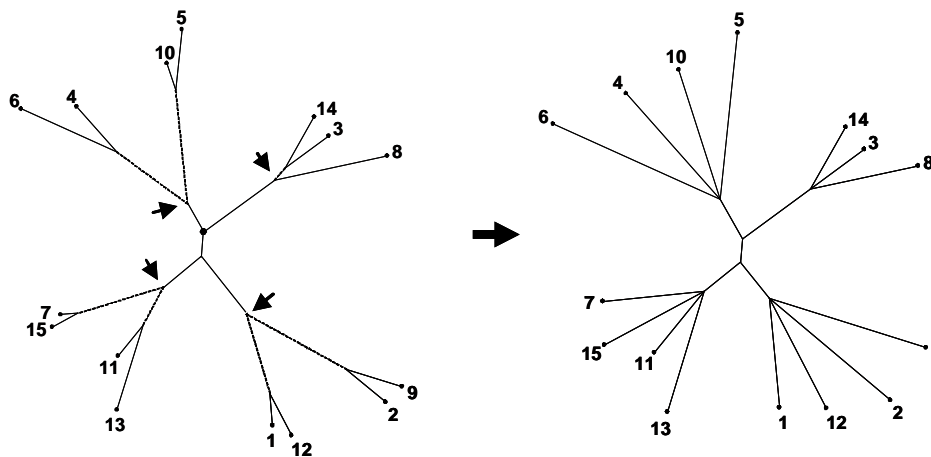
With an appropriate sequence of selecting / unselecting actions, it is possible to contract any internal edge subset.

- Right clicking on a node is equivalent to simultaneous left clicks on all nodes that are at an equal or greater distance from the current root. It is an automatic mode that contracts all edges beyond a user defined level.

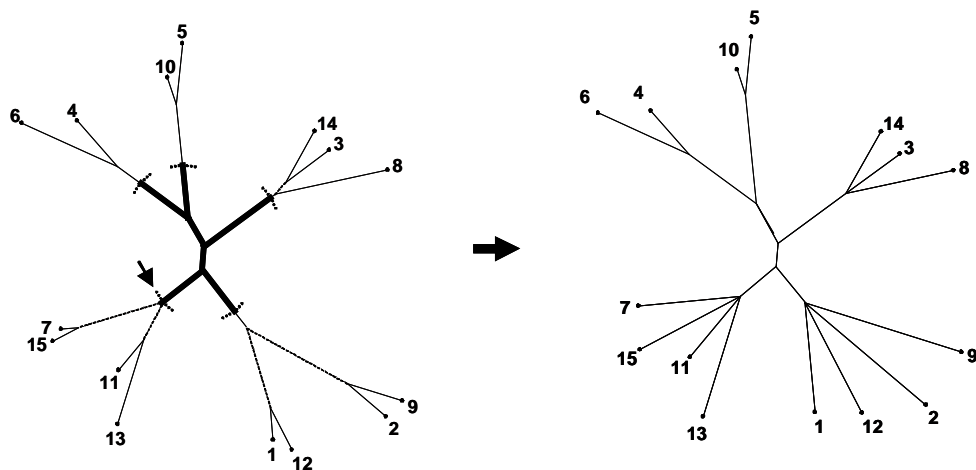


Open a window to record the contracted tree in a new .ARB file. A new tree draw window is automatically opened for this tree.

Examples:



contractions (dashed lines), arrows: user selected nodes



contractions (dashed lines), automatic mode, arrow: user selected threshold

## Group definition

This function partitions the unit set in groups, a group gathering all units of the sub-tree rooted on a selected node. A unit not implied in a sub-tree defines its own group. A group is identified by the node number or by the unit number for single unit groups. This group identifier affected to each unit is recorded in a new or a pre-existing .DON file. This group identifier will be used as external identifier in other analysis.



Button in the toolbar: toggle tool



- Left clicking on a node (or selecting it in the node list) affects all units of the sub-tree rooted on this node (relatively to the current root of the tree) to a same group, the edges of this sub-tree appear as bold lines.
- Left clicking on a node previously selected undoes the action. Selecting a sub-tree that includes previously defined groups, merges all concerned units in a same group.

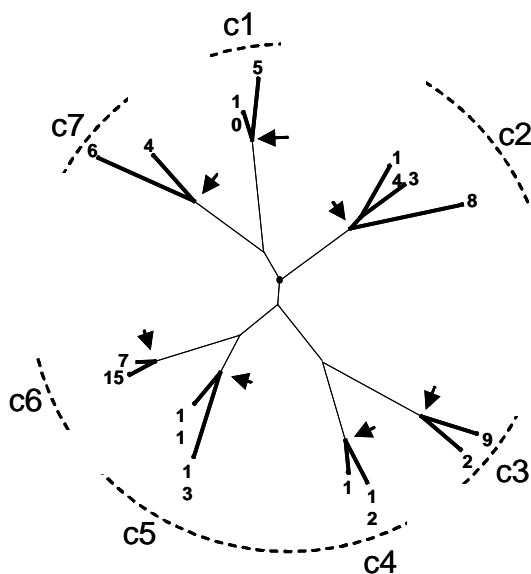
- Right clicking on a node is equivalent to simultaneous left clicks on all nodes at an equal or greater distance from the root.



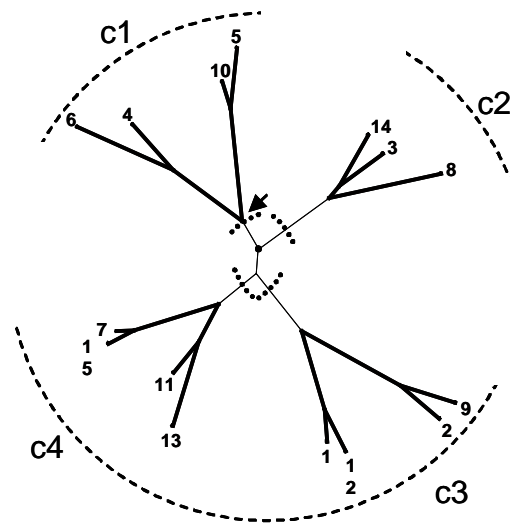
Open a window to define a .DON file where an identifier will be created to record the group numbers:

- If a .DON file has been previously opened for tree illustration, it is proposed to add the group identifier to this file
- It is also possible to select another existing identifier file or to create a new file.

Examples:



Groups (bold lines),  
arrows: user selected nodes



Groups (bold lines), automatic mode  
arrow: user selected threshold



*This window is multi instantiable and can be opened several times.*

## Edge length bargraph

This graphical tool displays the distribution of the edge lengths in a tree.

**Input file:** a .ARB file

### Graphical window

Bargraph parameters can be modified by the user:

- **Start value:** lower limit of the first class.
- **Class number**
- **Step:** class range

Any value lower than the start value is joined to the first class. Any value greater than the superior limit of the last class is joined to this last class.



**Copy** graph to clipboard:

Left click: EMF (vector) format

Right click: BMP (raster) format



**Print** the graph on the current printer



***This window is multi instantiable and can be opened several times.***

## ***Tree distance***

Additive tree distances between units two by two are calculated and stored in a dissimilarity matrix

**Input file:** a .ARB file

**Output file:** a .DIS file

By default, the proposed filename is *name\_Tree.DIS* where *name* is the initial .VAR file name.

The comment field in this file stores the tree file name and its characteristics.

**Output window**

Display the resulting dissimilarity matrix if 'Display when done' is checked.

## ***Fit criterion***

This function calculates some criteria to evaluate the fit between the initial dissimilarities  $d$  and the distances  $\delta$  as represented in a tree.

**Input files**

- a .ARB file for distances  $\delta$  as represented in the tree
- a .DIS file for initial dissimilarities  $d$ . The .DIS file read in the tree file comment field is automatically proposed but another .DIS file can be selected.

**Text window**

- mean error:  $\frac{1}{n} \sum_{i,j} (\delta(i,j) - d(i,j))$
- mean absolute error:  $\frac{1}{n} \sum_{i,j} |\delta(i,j) - d(i,j)|$
- maximum absolute error:  $\max |\delta(i,j) - d(i,j)|$
- mean square error:  $\frac{1}{n} \sum_{i,j} (\delta(i,j) - d(i,j))^2$



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## ***Least-squares re-estimation of edge lengths***

### **Method**

Whatever may be the tree construction method used, the edge lengths estimated at each step are not overall optima for the whole tree. For example, in Neighbor-Joining algorithm, these edge lengths are only local mean-squares estimators (for the partial structure defined at the current step) but are not optimal estimators when considering simultaneously all the units and their structure in the final tree. Note that these estimated edge lengths are not involved in the subsequent steps of the algorithms and the inferred tree structure will be the same what may be their estimations.

However when the tree topology is known, we are able to estimate edge lengths such that the distances in the tree are least-squares estimators of the initial dissimilarities. For that, the distance in the tree between two units  $i$  and  $j$  is expressed as the sum of the unknown edge lengths belonging to the path from  $i$  to  $j$  and this distance is expected to be the closest, in the least-squares sense, to the initial dissimilarity. This leads to solve a system of  $n(n-1)/2$  equations with as many unknowns as edges.

So starting from the tree produced by a tree construction algorithm, we can reestimate all edge lengths as least-squares estimators. This reestimation will cause loss of the ultrametric property.

### **Procedure**

#### **Input files:**

- a .ARB file for tree topology
- a .DIS file for dissimilarities. The .DIS file read in the tree file comment field is automatically proposed but another .DIS file can be selected.

**Output file:** a new .ARB file to record the tree with its new edge lengths. By default, the proposed filename is *name\_Adjusted.ARB* where *name* is the initial .ARB file name.

**Text window:** list the fit criterion (see [Fit criterion](#))

- for the initial lengths as read in the .ARB file in input
- for the re-estimated lengths



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

**Graphical window:** show the resulting tree if 'Display when done' is checked.



## Pruning

This function deletes one or several units in a tree. The corresponding external edges are removed and the resulting nodes of degree 2 are removed by merging the two corresponding edges.

**Input file:** a .ARB file

**Output file:** a new .ARB file to record the pruned tree. By default, the proposed filename is *name*\_Pruned.ARB where *name* is the initial .ARB file name.

### Selection of units to prune



Buttons to exchange part or all units between 'kept' or 'pruned' columns

- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

**Graphical window:** show the pruned tree if 'Display when done' is checked.

## Grafting

### Method

This function adds a new unit to a tree such that the tree distances between this new point and all the units in the tree are as close as possible of the corresponding initial dissimilarities.

The edge and the position of the grafting point on this edge have to be defined. The implemented algorithm examines successively each edge and estimates the position on the edge according to an optimality criterion. Finally, the edge which optimizes the criterion is retained.

Two optimality criteria are proposed. The first one is a least square criterion that might be used when the tree is inferred with Neighbor-Joining method. If  $x$  is the unit to graft,  $d$  the dissimilarity and  $\delta$  the additive tree distance, the position of the grafting point optimizes the quantity:

$$\frac{1}{n} \sum_i (\delta(i, x) - d(i, x))^2$$

The second criterion is analogue to the criterion used in Score method. Consider a triplet  $(i, j, k)$ , the unit  $x$  to graft and a given edge for grafting. The smallest of three sums of dissimilarities two by two for this quartet gives a first topology. The choice of an edge for grafting gives a tree topology for this quartet. If these two topologies are the same, the

score of the edge is set to 1. The resulting score for this edge is the sum for all triplets. Finally the edge giving the higher score is retained to graft the new point.

This function is particularly useful when an outgroup is added to the unit set in order to define a more ancestral node which will be used as root for the tree. However, outgroups by definition are relatively distant units and they often disturb the tree build on data including this outgroup. A better solution would be to graft *a posteriori* this outgroup on a tree build on the unit set (without the outgroup) and to regard the grafting point as the ancestral root.

## Procedure

### Input files:

- a .ARB file
- a .DIS file for dissimilarities. The .DIS file read in the tree file comment field is automatically proposed but another .DIS file can be selected.

### Selection of the unit to graft

- Select the unit to graft in the list of available units in the left column.
- [optional] select a .DON file and an identifier in this .DON file to facilitate unit selection.

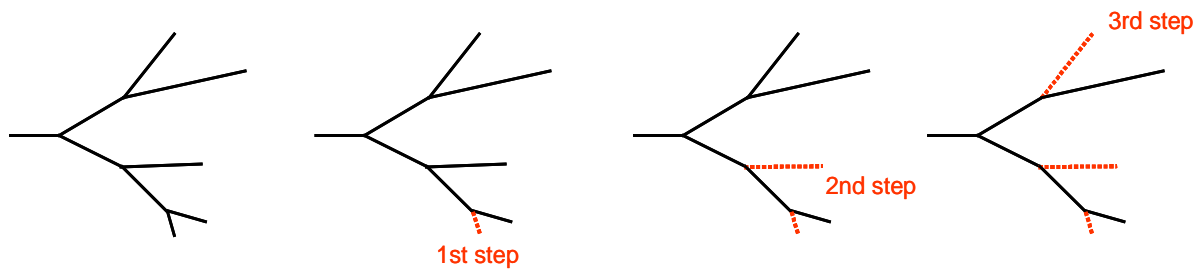
**Output file:** a new .ARB file to record the resulting tree. By default, the proposed filename is *name\_Grafted.ARB* where *name* is the initial .ARB file name.

**Graphical window:** show the grafted tree if 'Display when done' is checked, the new edge is in red.

## ***Max length sub tree***

### Method

Starting from a set of units, this procedure searches for a subset of units minimizing the redundancy between units and limiting if possible the loss of diversity. Redundancy means that some units are very close and then bring in part the same information on the diversity. The diversity is here expressed by the tree as build on the initial set of units. Two units are regarded as redundant if their distance in the tree is small. The choice between these two units relies on their edge lengths. By construction, the unit with the longest edge has more uncommon characters than the unit with the smallest edge which shares characters with other units more frequently. So, in order to maintain at best the diversity in the tree, we choose to remove the unit with the smallest edge and to keep the unit with the longest edge.



The procedure is a stepwise procedure that proceeds by successive pruning of redundant units. Starting from a tree build with a convenient method, distances between units in the tree are calculated, the pair of units of minimal distance is selected and the unit with the smallest external edge is removed, the procedure iterates on this sub tree and so on while the sub tree has more than 3 units.

At each step, two indices are calculated to characterize the shape of the tree:

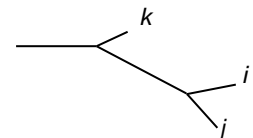
- the 'sphericity index' that is the ratio of the lengths of external edges to the total length of the current sub tree. This index has low values for trees with numerous redundant units (a lot of small external edges) and tends towards 1 when all units are independent (a lot of long external edges, the tree tends towards a star-like tree).
- the length of the pruned edge expressed in percent of the initial tree length,

The procedure lets to the user the choice of the sample size to retain. For a selected size, the sample composition and the corresponding sub tree can be recorded.

(This procedure is also used in a particular genetic background, see [sampling for disequilibria](#))



*This procedure is not equivalent to remove at each step the unit with the smallest edge. This would not be efficient to reduce redundancy, ex: i and j are the redundant units although k has the smallest edge.*



## Procedure

### Input file

A .ARB file in input.

### Units

A particular status can be defined for each unit:

- **Excluded** units: these units are definitively discarded and the procedure starts from a tree where these units are pruned,
- **Forced** units: these units will be retained in any case in the sub tree. For a selected pair of minimal distance, if one unit is forced, it will be kept whatever the edge lengths, if the two units are forced, the two are kept and another pair is selected,

- **Removable** units: these units are neither excluded nor forced and are available for selection in the procedure.

(excluded + removable + forced = **initial** unit set, removable + forced = **active** unit set)

The currently excluded/removable/forced units are listed in the three columns.



Buttons to move part or all units between adjacent columns

- **Identifiers** to select a .DON file and an identifier in this .DON file if selection is easier using another unit identifier.

Remark: the current selected identifier will be used in the text window to identify units.

## Graphical window

This graph displays for successive decreasing sample size:

- 'sphericity index'
- ratio pruned edge on initial tree length.

The x-axis is in number of units in the current sub tree:

- The first point on the left (step 0) is for the tree on the active units (or the initial units if there is no excluded unit),
- The last point on the right is for the tree of user-defined size (see '**Unit number**' value below). By default this value is the minimal size for a sub tree: the number of forced units or 3, the three last units, if there is no forced unit.



**Copy** graph to clipboard:

Left click: EMF (vector) format

Right click: BMP (raster) format

## Text window

A table with in line, for each step:

- **Step**: current step,
- **Sub tree size**: the number of remaining units at this step,
- **Removed unit number**: the unit removed at this step identified by its numerical value,
- **Removed unit identifier**: the unit removed at this step identified by the last selected identifier in Units sub menu,
- **Removed edge**: the length of the removed edge,
- **Current tree length**: the total length of the sub tree at this step,
- **External edges**: the total length of external edges at this step,
- **Sphericity**: the sphericity index,

A first line in red is for the initial tree on the whole data set. The line in blue is for the step 0 on the active unit set.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Sub tree

### • Unit number

The user chooses the retained sub tree by its size. It is a value between the number of active units (without the excluded units) and the number of forced units or 3, the three last units, if there is no forced unit.

### • Identifier file

A .DON file is selected to record the results of the procedure for each unit. It may be an existing .DON file where all units of the .ARB file in input are necessarily present, the new fields will be added after the fields already existing. It may be also a new .DON file which is initialized with the units of the input .ARB file and if an identifier has been selected in **Units** sub menu, the first recorded field is this unit identifier.

### • Record Identifier

- The first recorded field is the step at which the unit has been removed (forced units have all the value of the last step). For excluded units the value is 0. This identifier is independent of the selected sample size and can be used to characterize any subset: for a size  $m$ , all units with a value lower or equal to  $m$  are in the subset. The label of this field is by default 'Tree remove order' but it can be modified by the user.

- The second recorded field is proper to the user selected sub tree and takes values:

- 'Excluded' if the unit was excluded (value 0 in the previous field)
- 'Removed' if the unit was removable and has been effectively removed,
- 'Kept' if the unit was removable but has been kept in this sub tree,
- 'Forced' if the unit was forced (kept in any case in the sub tree).

The label of this identifier is by default 'Tree sample\_m' for a sub tree of size  $m$ , but it can be modified by the user.

If several subsets of different size are recorded, 'Tree remove order' field will not be repeated.

### • Record subtree

Ask for the name of a new .ARB file to record the sub tree corresponding to the selected '**Unit number**' value, and open automatically a window to display this tree.

## ***Add a 2-degree node***

This function adds a node of degree 2 on a user-defined edge of a tree. This node creates two new edges of equal length. This function is used for example for a more convenient representation in rooting the tree on an ancestral node created on the edge between an outgroup and the other units. It is used also to define a common root to two trees compared with the [rooted MAST](#) procedure.

**Input file:** a .ARB file

**Output file:** a new .ARB file to record the resulting tree. By default, the proposed filename is *name\_2-Node.ARB* where *name* is the initial .ARB file name.

### **Edge selection**

. Select an edge in the 'Edges list'

**Graphical window:** show the resulting tree if 'Display when done' is selected

## ***Remove all 2-degree nodes***

This function removes all nodes of degree 2 in a tree by merging the two connected edges.

**Input file:** a .ARB file. If selected has no 2-degree node, input file procedure fails with a user message.

**Output file:** a new .ARB file to record the resulting tree. By default, the proposed filename is *name\_R2-Node.ARB* where *name* is the initial .ARB file name.

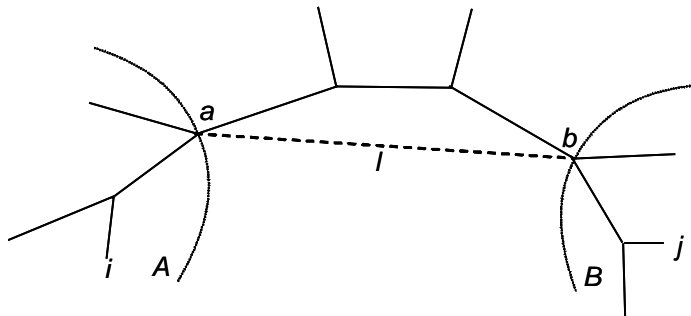
**Graphical window:** show the resulting tree if 'Display when done' is checked.

## ***Reticulations***

### **Method**

We consider a dissimilarity  $\delta$  on a set of  $n$  units and  $T$  the associated tree. This tree can be interpreted as a phylogenetic tree only under assuming a model of evolution by accumulation of inherited mutations excluding any form of genetic exchange between separated branches. This model is valid in case of asexual multiplication or for mitochondria or chloroplast markers that have single parent heredity. It is still usable when analysed taxa are geographically isolated and cannot exchange. In other cases it may produce partially false results, the diversity structure being a network and not a tree.

Several attempts to represent diversity by networks can be found in the literature. The most popular is the 'split-decomposition' (logiciel SplitTree, Bandelt and Dress) but which in practice produces unreadable graphs with a huge number of edges when the number of units exceeds a few tens. To recover a lower complexity, a way, proposed by Legendre and Makarenkov (2004), is to start with a tree and to improve this tree by addition of reticulations when necessary. This model assumes that the tree model is correct on the whole but is locally disturbed by some rare events of horizontal exchange.



A variant of this method is implemented in DARwin. Let  $A$  (and  $B$ ) be the subset of  $n_A$  (and  $n_B$ ) units having  $a$  (and  $b$ ) as common ancestor. If we assume some genetic exchange between these ancestors  $a$  and  $b$ , the observed dissimilarity  $\delta_{ij}$  between a unit  $i$  of  $A$  and a unit  $j$  of  $B$  is probably lower than the dissimilarity  $d_{ij}$  in the tree which is conditional to all other units for which the tree like process is valid. Then the representation will be improved if a shortest edge of length  $l$  is added between  $a$  and  $b$ . The path between two units can now pass by this new edge if a unit is in  $A$  and the other one in  $B$  or by the tree path in all the other cases.

The fit between the data and the inferred tree can be estimated by the quadratic sum of differences between the observed dissimilarity  $\delta$  and the tree distance  $d$ .

The initial quadratic sum  $W_0$  is:

$$W_0 = \sum_i \sum_{j \neq i} (\delta_{ij} - d_{ij})^2$$

with  $d_{ij} = d_{ia} + d_{ab} + d_{bj}$

When the reticulation ( $ab$ ) is added, the quadratic sum becomes  $W_{ab}$ , the distance in the tree for a unit  $i$  in  $A$  and a unit  $j$  in  $B$  being taken as:

$$d_{ij} = d_{ia} + l + d_{bj}$$

keeping fixed the lengths of all the other tree edges.

The gain in fit induced by the reticulation ( $ab$ ) is given by a criterion  $Q_{ab}$ :

$$Q_{ab} = \frac{W_0 - W_{ab}}{W_0}$$


and the length  $l$  is chosen to maximize this criterion:

$$l = d_{ab} + \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} (\delta_{ij} - d_{ij})$$

with the constraint  $0 \leq l \leq d_{ab}$

The gain in fit is successively computed over all pairs of nodes (internal and external nodes) providing an ordered list of all the potential reticulations. The best gain is retained for the first reticulation (the four higher values are also displayed to detect possible almost so good solutions). The distances in the tree for pairs of units in  $A$  and  $B$  are updated to account for the shortcut passing by this first reticulation and the algorithm iteratively detects the following reticulations.

A predefined least-squares threshold  $q$  can be used to retain only reticulations with a  $Q$  value greater than this threshold.

 *Warning: the edge lengths of the tree must be previously corrected by a posterior overall estimation ([Least-squares re-estimation of edge lengths](#)). Tree algorithms like NJ estimate the edge lengths at each step as local least-squares optima for the current step; that does not ensure that they are globally optimal. So without posterior overall estimation, reticulations will mainly try to correct this lack of optimality instead of accounting for genetic exchange between divergent branches.*

## Procedure

### Input files:

- a .ARB file for tree topology
- a .DIS file for dissimilarities. The .DIS file read in the tree file comment field is automatically proposed but another .DIS file can be selected.

### Number of reticulations

Stop the search when this maximal number of reticulations is reached.

### LS threshold %

Stop the search when the least-squares gain  $Q$  becomes lower than this threshold.

(these two parameters are combined and the search stops as soon as one of these parameters is reached)

Output file: a new .ARB file to record the tree with added reticulations. By default, the proposed filename is *name\_Reticulated.ARB* where *name* is the initial .ARB file name.



Text window:

- $W_0$  the sum of squares for the initial tree
- for each successive reticulation, by decreasing order of adjustment improvement:
  - the two nodes linked by the reticulation
  - $DW$ , the decrease in sum of squares  $W_0 - W_{ab}$  induced by this reticulation
  - $DW/W_0\%$ , this decrease in percent of the initial sum of squares
  - the same information for the 4 following best solutions at this step



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

**Graphical window:** show the resulting tree if 'Display when done' is checked. The reticulations are illustrated by dotted red lines joining the nodes but the tree geometry is preserved and the reticulations cannot be displayed at their true lengths.

---

# Tree comparison

## ***Consensus and tree distances***

### **Consensus methods**

The same unit set is often characterized by several variable sets: molecular markers on nuclear and mitochondrial DNA, several gene sequences ... and a common analysis of these data is logically a question of interest. A first solution would be to merge the different variable sets in a unique set and to construct a tree on this set. But it is rarely the right solution, for theoretical reasons when the different sets have not undergone the same evolutionary process or for practical reasons, for example when several gene sequences of very different lengths are compared, a direct concatenation would give undesirable higher contribution to the longest genes. A better solution is to construct a tree for each variable set and in a second time, to exhibit eventual common structure between these trees.

Consensus methods construct a synthetic tree which exhibits the common information to compared trees in retaining only the edges that are present in all elementary trees (or a majority). It is a purely structural point of view since edges lengths that depend on each particular variable set, are not directly comparable.

DARwin proposes the strict and the majority rule methods. The strict consensus only retains the edges which are present in all the trees compared. The majority rule consensus is less restrictive and retains the edges present in more than T% of the trees where T=50 is often chosen as threshold but could be greater. If only two trees are compared, the two methods are equivalent.

The edge lengths in the consensus tree are arbitrary set to 1. However, if some edges have null length in the initial trees, the length in the consensus tree depends on the chosen method:

- for strict consensus: if an edge has a null value in a least one tree, then this edge is null in the consensus
- for majority rule consensus: a consensus edge is null only if this edge is null in more than T% of the trees.



*Nodes of degree 2 create two edges which are equivalent in terms of bipartition and which do not contribute to the consensus construction, so they are ignored in the procedure.*

### 'Bipartition' distance between trees

In complement to the common structure revealed by the consensus tree, the difference between two trees can be summarized by a distance measure between these trees. This distance can be defined as the sum of differences of each one to their consensus tree which is always simpler in structure than the initial trees. This structure is quantified by an index of complexity  $v$ ; the dissimilarity between two trees  $H$  and  $H'$ , of consensus  $H_c$ , is thus:

$$d(H, H') = v(H) + v(H') - 2v(H_c)$$

Several definitions of complexity can be considered. A common way is to measure the complexity of a tree by its number of edges. The '**edge' distance** (=Robinson and Foulds distance) is thus the number of edges which are present in one tree and not in the other one and varies according to the number of internal edges conserved in the consensus tree. This distance regards all edges equivalently wherever their position in the tree may be. Another way is to weight the edges according to their ability to 'structure' the tree. These weights are defined from the corresponding bipartitions. If an edge induces a bipartition of  $s$  units against  $n-s$  units, the weight for this edge will be  $s(n-s)$ . So the weights will be the smallest for an external edge ( $n-1$ ) and maximal for a more 'structuring' internal edge partitioning the units in two equal subsets ( $n^2/4$ ). Then the complexity of a tree becomes:

$$v(H) = \sum_E s(n-s)$$

where  $E$  is the set of all edges of the tree.

The '**bipartition' distance** is then defined from the complexities according to the previous equation.

The complexity is normalized by the maximum value corresponding to a string-like tree, so the distance keeps a variation between 0 and 1.



*The bipartition distance depends on the consensus complexity. If several trees are concerned, we retain the consensus of all these trees. So the distance calculated between two trees will not be the same if only these two trees are compared or if other trees are implied in the consensus.*

A statistical interpretation of this distance requires its distribution under a null hypothesis that the trees are randomly drawn in the set of all possible trees on  $n$  units. This distribution is not analytically defined so we can only propose results from large simulations for binary trees. The following table gives for trees of 20 to 100 units the  $D_b$  value such that, among 6 000 random tree pairs,  $p\%$  show a distance lower or equal to  $D_q$ . This means for example that, for 100 units, a  $D_b$  lower than .412 has only 5 chances on 100 to be a random effect and only 1 chance if it is lower than .394.

$n$	1%	5%	10%	20%
20	.727	.753	.766	.782
40	.580	.600	.611	.626
60	.492	.511	.522	.537
80	.435	.453	.464	.477
100	.394	.412	.421	.434

## Procedure

**Input files:** several .ARB files (necessarily on the same unit set).



to add new tree files



to remove highlighted trees

**Output file:** a .ARB file which stores the consensus tree.

Strict or Majority rule consensus and the threshold for majority rule.

## Text window

- List of edges and their length in the consensus tree
- The 'bipartition' complexity of each tree
- The 'bipartition' complexity of the consensus
- The matrix of 'bipartition' distances between trees two by two.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

**Graphical window:** show the consensus tree if 'Display when done' is checked.

## ***Maximum agreement sub-tree (MAST)***

### Method

Consensus methods suppose that all the units are correctly represented; the exchange of an unit from one group to another is a strong indication that there is no real separation of the groups. But in some cases, the compared trees seem to have a common structure for almost all units except some ones which seem more erratic. For consensus trees, these erratic units have the same weight as others and may mask a common structure. Another point of view would be to identify and eliminate these 'fluctuating' units to exhibit the common structure. The problem becomes that of determining the smallest set of units that have to be pruned in each tree to obtain identical trees or, inversely, the

largest set of units having the same structure in the compared trees. These units form the maximum agreement sub-tree.

The simple statement of the problem masks a relatively complex algorithmic problem. DARwin uses an algorithm published by Kubicka et al. (1995) which gives an exact solution with a sufficiently low complexity to be used in practice. The algorithm relies on the enumeration of all possible solutions while limiting the depth of exploration of the branches on a stop criterion.

The two compared trees can be either unrooted or rooted. A tree is regarded as rooted if it has a node of degree 2 which will be taken as root (see [Add a 2-degree node](#)). If the two trees are rooted, a rooted MAST is available; it is more restrictive than unrooted MAST since common sub-structures have to respect the root position.



*This procedure concerns only two binary trees.*

## MAST order as tree distance

The order  $o$  of the MAST (the number of units conserved) can be considered as a measure of the resemblance between trees. The maximum order is  $n$ , and it is obtained for two identical trees, the minimal value is 3, since the single typology of three points is necessarily common to two trees.

So a distance between trees is defined as  $d = \frac{n-o}{n-3}$  ( $0 \leq d \leq 1$ )

The practical use of this criterion requires knowledge of the distribution of  $o$  under the hypothesis of independence of trees. This distribution is not known but has been approached by simulating pairs of random binary trees of 20 to 100 leaves. The above table gives the proportion of random tree pairs found for a given order and the average order for all the simulated pairs.

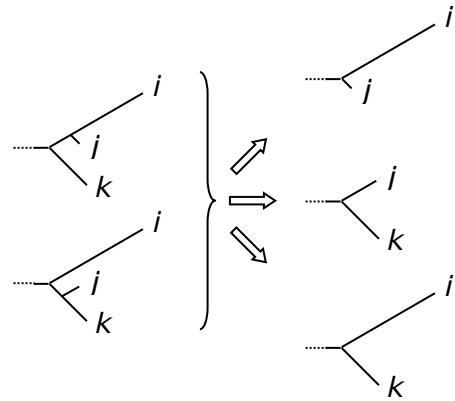
$o=$	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Mean
$n=20$	.01	.341	.497	.139	<b>.012</b>	.001										7.80
$n=40$				.039	.368	.405	.157	<b>.027</b>	.004							10.78
$n=60$							.275	.427	.201	.063	<b>.012</b>					13.04
$n=80$								.061	.296	.369	.199	.059	<b>.007</b>			14.95
$n=100$									.009	.163	.359	.297	.129	<b>.037</b>	0.06	16.51

For example, for 60 units, random trees will have in average an order of 13 and an order of 15 will appear in frequency of 0.063. This can be used to fix a rule of interpretation. For example, it can be concluded that orders superior or equal to 10, 13, 16, 18 and 19 (when  $n$  varies from 20 to 100) will be a random effect in less than 5 cases on 100.

## Maximal length MAST

In general, the maximal order is obtained for several trees corresponding to different unit subsets and only the first found tree is retained as solution. However, it could be considered that trees implying more typical units are more representative trees, typical units meaning units with the longer external edges.

In this example, the two topologies for  $(i,j,k)$  give



three equivalent solutions in terms of MAST. If edges lengths are used to decide between these solutions, then the third one which keeps  $i$  and  $k$  is retained since their edges are the longest. So between all solutions of maximal order, the algorithm implemented in DARwin retains the tree of maximal total length. This requires estimating the length of each edge in the solution tree from the lengths of these edges in the two initial trees. This estimate is the weighted average of the two initial lengths, with a user-defined weight for each tree:

$$l = (w_1 l_1 + w_2 l_2) / (w_1 + w_2)$$

## Procedure

**Input files:** two .ARB files on the same unit set.

Trees with nodes of degree greater than 3 are not allowed.

If the two trees have a 2-degree node, the two trees are assumed rooted on these nodes.

If only one tree has a 2-degree node, the two trees are assumed unrooted and the 2-degree node will be ignored after user confirmation.

**Output files:** a new .ARB file which stores the resulting tree.

For ordinary MAST, all edges are arbitrary set to 1.

For Maximal length MAST, the edge lengths are the weighted averages of the initial lengths.

## Parameters

- **ordinary or Maximal length MAST + Weights** for edge length estimation:
  - Ordinary MAST all edges are set to 1
  - Maximal length MAST:
 
$$l = (w_1 l_1 + w_2 l_2) / (w_1 + w_2)$$
 with particular cases:
    - $w_1 = w_2 (\neq 0)$  then  $l = (l_1 + l_2) / 2$
    - $w_1 = 0$  and  $w_2 \neq 0$  then  $l = l_2$  or conversely
    - but  $w_1 = 0$  and  $w_2 = 0$  is not allowed

- **rooted or unrooted MAST** (only if the two trees have a 2-degree node)

### Text window

- Characteristics of the two input trees
- Ordinary or maximal length MAST + weights defined for each tree
- uprooted or rooted trees and in this case the root position in each tree
- Order of the maximum agreement subtree



**Print** the text window on the current printer



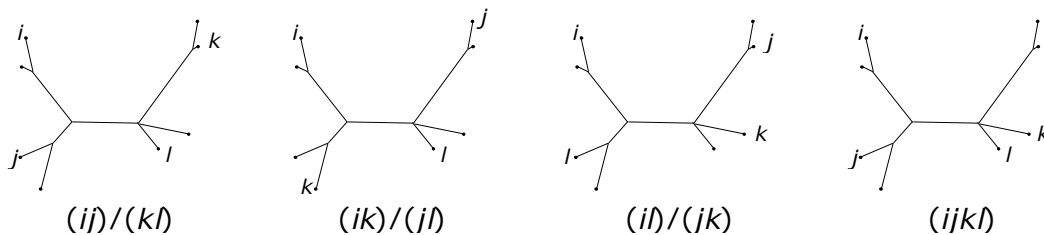
**Save** the text window in a .TXT or .RTF file.

**Graphical window:** show the resulting tree if 'Display when done' is checked.

## Quartet distance

### Method

The quartet distance estimates the difference between two trees as the number of quartets which have not the same topology in the two trees. It is a purely topological measure that does not depend on the edge lengths.



Four topologies are possible for a quartet  $(i, j, k, l)$ , the three first are resolved topologies with two units against the two others. The fourth is an unresolved topology which implies a node of degree greater than 3.



*Some algorithms never infer nodes of degree greater than 3 but some edges may have a null length. The implemented quartet distance algorithm regards that as an artefact and virtually contracts these edges of null length in creating virtual nodes of degree greater than 3.*

Let  $N_q$  be the total quartet number,  $N_r$  the number of resolved quartets of same topology in the two trees and  $N_u$  the number of quartets which are unresolved simultaneously in the two trees, then the quartet distance  $D_q$  is:

$$D_q = (1 - N_r - N_u) / N_q$$

A direct counting for all quartets would lead to a complexity in  $o(n^4)$ . The implemented algorithm based on a topological tree decomposition allows a complexity in  $o(n^3)$ . However this method may be very long for large data sets.

This quartet distance is only a measure of dissimilarity between two trees. For a statistical interpretation of this value, it would be necessary to know the distribution of this measure under the null hypothesis that the two trees are independent trees randomly drawn in the population of all trees. This distribution is not known so we can only propose results from large simulations for binary trees. The following table gives for trees of 20 to 100 units the  $D_q$  value such that, among 6 000 random tree pairs, p% show a distance lower or equal to  $D_q$ . This means for example that, for 100 units, a  $D_q$  lower than .654 has only 5 chances on 100 to be a random effect and only 1 chance if it is lower than .646.

$n$	1%	5%	10%	20%
20	.552	.598	.619	.641
40	.611	.634	.643	.655
60	.629	.646	.652	.659
80	.640	.651	.656	.661
100	.646	.654	.658	.662



*This algorithm can be time consuming with large trees (over 2 hundred units).*

## Procedure

**Input files:** Two .ARB files (necessarily on the same unit set).

**Text window:**

- For each tree, resolved and unresolved quartet numbers
- Total number of quartets
- Number resolved in the two trees with the same topology
- Number of quartets resolved with different topologies
- Number of quartets resolved in a tree and unresolved in the other one
- Number of quartets unresolved in the two trees
- Quartet distance



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.



# Disequilibrium

## ***Background***

This submenu addresses a specific problem of linkage disequilibrium in genetic. The background is the localization of genes involved in agronomic traits in using molecular markers as tags. The closeness between gene and marker is detected by association mapping based on disequilibrium between linked loci. For that, large germplasm collections are often used in order to maximise the allelic diversity. But collections are often complex pools of genetically differentiated objects accumulating various demographic or breeding events. So these highly structured populations show disturbed balance of alleles generating disequilibria even between unlinked loci leading to false linkage disequilibria.

For association mapping analysis, a large number of accessions are genotyped for molecular markers but phenotyping for phenotypic characters is long and expensive and can concern only a small part of the collection. So combining the two problems: - spurious associations in heterogeneous populations - sampling for phenotyping, arises to the question: how to take advantage of the necessary sampling to minimize spurious associations due to structures, if possible with a limited allelic diversity reduction?

The observed disequilibrium is the sum of a physical component due to linkage between loci on the chromosome (LD) and a structural component due to structures in the population (SD). The sampling procedures will work on a set of independent markers in order to remove the linkage component.

## ***Method***

The proposed methods are heuristic procedures that find approximate solutions since exact solutions should examine all possible subsets and are not feasible for large populations. Two different heuristics are proposed. The first one starts from a diversity tree on a set of unlinked makers and extracts a sub tree as unstructured as possible. The second one starts from the disequilibria observed between loci and extracts a sample minimizing these disequilibria.

### **Max length sub tree**

For the first approach on diversity tree, it is assumed that structures in the population can be viewed in any case as overrepresentation of some groups, inducing large redundancy between units (=accessions). It is expected that the deletion of redundant units will reduce the general disequilibrium. So the procedure searches for the most

unstructured tree -a star like tree- by successive pruning of redundant units, this sub tree being of maximal length to maintain a sufficient allelic diversity. At each step, disequilibria between pairs of loci are calculated to follow effect of redundancy decrease on disequilibria.

This procedure is a development in a particular background of a general procedure to prune a tree of its redundant units (see [Max length sub tree](#)).

### Min SD subset

The second approach relies directly on the disequilibrium between loci observed on the data set and tries to find a sub sample that shows the smallest disequilibrium. The procedure is a stepwise algorithm removing at each step the unit with the greatest contribution to the general disequilibrium between all pairs of loci.



*These two procedures do not lead to the same result. The second one, on disequilibria, gives logically best results in term of disequilibrium decrease but it might be an ad hoc solution and the result on an other set of markers is questionable. The sub tree procedure is often less efficient to reduce disequilibria but it might give more robust samples. However the two procedures are not exclusive. On real data, we applied with success a mixed strategy: max sub tree was used in a first step to extract a sub sample where the most redundant units were removed and in a second step, Min SD procedure was used on this sub sample to effectively minimize the disequilibrium.*

### Haplotypes

We consider here only diploid species. Disequilibrium between two loci can be estimated only if we can say which allele of a locus is associated to which allele of an other locus. So it is assumed that the phases are known and that the two haplotypes of a heterozygous diploid are identified.

In the absence of family data and molecular-haplotyping methods, statistical methods are required for inferring haplotypes from genotypic data. Several software are available, they often implement versions of the maximum-likelihood expectation/maximization algorithm as proposed by Excoffier and Slatkin (1995). To contend with some limitations of these algorithms, Stephens et al. (2001) propose a Bayesian approach using a Monte-Carlo Markov chain algorithm. The procedure is implemented in the software PHASE, available from <http://www.stat.washington.edu/stephens/software.html>. Two modules are proposed to export DARwin files to Phase and to extract DARwin files from Phase outputs.

### Linkage versus structure disequilibrium

For diallelic loci, several measures have been proposed for linkage disequilibrium but the Lewontin's  $D'$  is known for its good properties. It uses for standardization the maximum value of disequilibrium observed at the date of the mutation before any recombination. Linkage disequilibrium is fundamentally a balance between cells of the contingency table,

allelic frequencies, the margins, are fixed and have not changed since the date of mutation.

Example:

		B		
		1	2	
A	1	24	46	70
	2	16	14	30
		40	60	100

$$d = 24 \times 14 - 46 \times 16 = 4 \times 100$$

. equilibrium

24+4	46-4	70
16-4	14+4	30
40	60	100

. maximum

10	60	70
30	0	30
40	60	100

$$d_{\max} = 60 \times 30 = 1800$$

$$D' = \frac{400}{1800} = 0.22$$

This measure can be extended to multiallelic loci, it is a weighted sum of  $D'$  calculated on all possible 2x2 sub tables crossing an allele at a locus and the sum of all other ones:

$$D' = \sum_{i=1}^K \sum_{j=1}^L p_i p_j D'_{ij} \quad \text{with } D'_{ij} \text{ calculated on}$$

		i	non-i	
j				$p_j$
		non-j		
				$p_i$

A rational for this extension is that disequilibrium depends only on the distance between the two loci and consequently the same value should be obtained what were the alleles considered.

But this measure of linkage disequilibrium cannot be used directly for structure disequilibrium essentially and clearly because they have not the same origin. A first question is the standardization. Disequilibrium is no longer a balance between cells and the margins do not represent in any way the initial population. So the only possible standardization term is the maximum value that occurs in the limit case with an half of the population in two opposite cells of the contingency table:

		B		
		1	2	
A	1	24	46	70
	2	16	14	30
		40	60	100

$$d = 24 \times 14 - 46 \times 16 = 4 \times 100$$

. nearest equilibrium

24	46	70
16-7	14	23
33	60	100

. maximum

50	0	50
0	50	50
50	50	100

$$d_{\max} = 50 \times 50 = 2500$$

$$D' = \frac{400}{2500} = 0.16$$

A second question is the extension to multiallelic loci. Disequilibria are no longer proper to the loci themselves but different pairs of alleles at a same locus may exhibit different

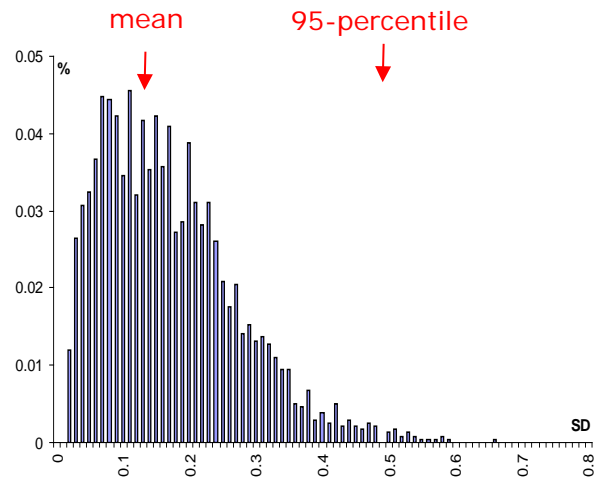
levels of disequilibrium. So we propose a new measure as a sum on all possible pairs of alleles:

$$SD = \frac{1}{D_{\max}} \sum_i \sum_{i' \neq i} \sum_j \sum_{j' \neq j} (n_{ij} n_{i'j'} - n_{ij'} n_{i'j})$$

with  $D_{\max} = \frac{1}{2} NA(NA-1) \left( \frac{n}{NA} \right)^2$  where  $NA = \min(K, L)$  and  $K$  and  $L$ , the numbers of alleles at the two loci.

### Disequilibria on a set of loci

A disequilibrium is calculated for each pairs of loci ( $L(L-1)/2$  values for  $L$  loci) and a population is characterized by the distribution of these disequilibria. A first synthetic criterion is the general mean; however distributions are often asymmetric with a lot of small values (loci in equilibrium) and some larger values (loci in disequilibrium). If a sample reduces effectively the disequilibrium, it is essentially in reducing these greatest values. So a better criterion to reflect this queue of distribution is an upper percentile (to a fixed threshold).



### Random samples as reference

To evaluate effect of sampling on disequilibrium decrease we need for a reference. As disequilibrium depends on the sample size, this reference cannot be the initial estimation on the whole population. A better reference is the disequilibrium observed on samples of same size randomly drawn in the initial population. For that, a large number of samples of a given size are randomly drawn, for each one the mean and the percentile of the SD distribution are calculated. The means of these two criteria on all the random samples serve as reference for the sample proposed by the sampling procedure.

### Allele richness

Two criteria inform on the loss of alleles in successive samples. The first one is the number of present alleles in a sample expressed in percent of the initial number of alleles. It can be considered also that the loss of a rare allele is less important than the

loss of a very frequent allele, so the second criterion takes into account the initial allele frequencies.

At a given step, let  $s_{lk}=1$  if allele  $l$  of marker  $k$  is present in the sample and 0 if it is absent:

$$C1 = \frac{1}{K} \sum_k \frac{1}{L} \sum_l s_{lk} \quad (C1=C2=1 \text{ at step 0})$$

$$C2 = \frac{1}{K} \sum_k \sum_l f_{lk} s_{lk}$$

where  $f_{lk}$  is the initial frequency of allele  $l$  for marker  $k$ ,  $K$  the number of loci,  $L$  the number of alleles at locus  $k$ .

### Algorithm for Max length subtree

It is a step by step procedure starting from a diversity tree inferred with a convenient method. Distances  $d(i,j)$  between units in the tree are firstly evaluated.

Then for each step, the pair of remaining units  $(i,j)$  of minimal distance is selected.

The unit  $i$  is removed if  $l_i < l_j$  and conversely,  $l_i$  and  $l_j$  being the lengths of the external edges in the tree.

The procedure iterates on this sub tree and so.

### Algorithm for Min SD subset

It is a stepwise algorithm removing at each step the unit with the greatest contribution to the disequilibrium. This contribution, the score, is evaluated for each unit and the unit with the highest score is removed.

For a pair of loci  $k$  and  $l$ , the partial score of unit  $i$  is the difference between the disequilibrium with and without this unit (:

$$X_i^{(kl)} = SD_{kl}^{+i} - SD_{kl}^{-i}$$

Then  $Sc_i$ , the score of unit  $i$ , is the weighted sum on all pairs of loci of its partial scores:

$$Sc_i = \sum_k \sum_l w_{kl} X_i^{(kl)}$$

The weight  $w_{kl}$  is chosen as the square of the disequilibrium for this pair of loci, to favour reduction of highest disequilibria:

$$w_{kl} = (SD_{kl}^{+i})^2$$

The unit of maximal score is removed, the procedure iterates on this sample and so on.

# Procedure

This presentation is common to the two procedures since inputs, outputs, options... are in great part the same. When necessary, specificities of each one (noted MLST for Max length sub tree and MSD for Min SD subset) will be described.

## Input files

. a .VAR file (allelic data type) on diploids with known haplotypes.

. a .ARB file for **Max length sub tree**

If a unit of the tree is not found in the allelic data file, the procedure stops and waits the user selects the right files.

If the allelic data file includes units that are not in the tree, these units are discarded and only units common to the two files are retained.

Three 'Tabs' allows user to set options. '**Options**' tab contains global options and 'OK' button to launch main procedure, '**Random sampling**' tab contains random sampling options, and '**Record sample**' tab contains record sample options.

## 'Options' tab

### Units

a particular status can be defined for each unit:

- **Excluded** units: these units are definitively discarded and will never be retained in samples,
- **Forced** units: these units will be retained in any case in samples,
- **Removable** units: these units are neither excluded nor forced and are available for selection in the procedure.

(excluded + removable + forced = **initial** unit set, removable + forced = **active** unit set)

The currently excluded/removable/forced units are listed in the three columns.



Buttons to move part or all units between adjacent columns

- **Identifiers** to select a .DON file and an identifier in this .DON file if selection is easier using another unit identifier.

Remark: the last selected identifier will be used in the text window to identify units.

- **Statistics on current selection** open a window listing some synthetic parameters for each unit (removable + forced): numerical identifier, selected identifier, status, number of missing alleles, of loci with at least one missing allele, of homozygote loci, of heterozygote loci.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Loci

Selection of a subset of loci.

The currently unselected/selected loci are listed in the left/right columns.



Buttons to move part or all loci between columns

- **Statistics on current selection** open a window listing synthetic parameters for each selected locus: locus identifier, number of missing units, number of alleles, min value and max value for allele code.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Missing data

Check the box to indicate that some data are missing in the allelic data file. The integer value retained to code a missing data has to be specified by the user (**code for missing data**).

Disequilibrium for a pair of loci will be calculated only on haplotypes with present data for the two loci.



*In case of frequent missing data, disequilibria between pairs of loci may be evaluated on very different numbers of units.*

**Min Freq** Minimal frequency for rare alleles in %

All alleles in number lower than this frequency by two times (two haplotypes for each unit) the size of the population at the current step will not be taken into account in disequilibrium estimation. Note that the number of units decreases at each step, so the threshold that depends on this number varies with steps.

## Percentile

Level for upper percentile of the SD distribution.

**OK** Launch the main procedure, display SD graph and the text window. It opens also options for random sampling, sample recording...

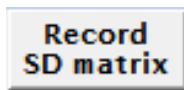


*Any modification in the previous options erases the graph and closes the subsequent options until the procedure is launched again.*



Display the distributions of disequilibria between pairs of loci, in blue for the active data set (selected loci and removable+forced units), in red for a given step selected by the user (**sample size**), this choice is proper to this window and is independent of the value used to record the sample (see below).

The number of classes, the starting value, and the class range can be modified.



Record the disequilibria between pairs of loci for the active data set (selected loci and removable+forced units) in a .DIS file on the number of selected loci. The associated .DON file is also created to record the labels of the loci. This matrix can be used to display a topology of loci according to their disequilibria (groups of loci in disequilibria) with a tree construction method on the transformed matrix 1-d.

### **'Random sampling' tab**

Options to create random samples (see [Random samples as reference](#)):

- **Step:** the estimations of disequilibria for a great number of random drawings can be long. It is not necessary to examine each sample size and a random sampling of sizes  $n$ ,  $n\text{-step}$ ,  $n\text{-}2\text{xstep}$ ... is often sufficient for interpretation.
- **Number of drawings:** number of random samples to draw for a given sample size.
- **Resampling:** runs random SD computing

### **Graphical window**

- **SD graph**

This graph displays for successive decreasing sample size:

- SD mean (red curve)
- SD percentile (green curve)
- Random SD mean (yellow points)
- Random SD percentile (blue points)

x-axis: in number of units in the current sample:

- the first point on the left (step 0) is for the sample on the active units (or the initial units if there is no excluded unit),
- the last point on the right is for the sample of user-defined size (see '**sample size**' value below). By default this value is the minimal size of a sample: the number of forced units or 3, the three last units, if there is no forced unit.

y-axis: SD x 100.



**Copy** graph to clipboard:

- Left click: EMF (vector) format
- Right click: BMP (raster) format

- **'tree' graph**

For **Max length sub tree**, another graph displays, with the same x-axis, the variations at each step of:

- 'sphericity index'
  - ratio pruned edge on initial tree length.
- (see [Max length sub tree – Method](#))



## Text window

A table that lists the main results, it differs in part for the two procedures.

### Text window for Max length sub tree

- **Step:** current step,
- **Sample size:** the number of remaining units at this step,
- **Removed unit number:** the unit removed at this step identified by its numerical value,
- **Removed unit identifier:** the unit removed at this step identified by the last selected identifier in Units sub menu,
- **Removed edge:** the length of the removed edge,
- **Current tree length:** the total length of the sub tree at this step,
- **External edges:** the total length of external edges at this step,
- **Sphericity:** the sphericity index,
- **SD mean:** the mean value on all pairs of loci,
- **SD percentile:** percentile of the distribution of SD on all pairs of loci,
- **Allele number:** number of alleles present in the current sample for the selected loci, all alleles are considered and the minimal frequency for rare alleles does not apply here,
- **Allele proportion:** (C1) the current number of alleles on the number of alleles at step 0 (on all removable and forced units),
- **Allele freq proportion:** (C2) the sum for alleles present in the current sample of their frequencies at step 0 (on all removable and forced units).

### Text window for Min SD subset

- **Step:** current step,
- **Sample size:** the number of remaining units at this step,
- **Removed unit number:** the unit removed at this step identified by its numerical value,
- **Removed unit identifier:** the unit removed at this step identified by the last selected identifier in Units sub menu,
- **SD mean:** the mean value on all pairs of loci,
- **SD percentile:** percentile of the distribution of SD on all pairs of loci,
- **SD max:** the maximal disequilibrium among all pairs of loci,
- **for locus pair:** the pair of loci for this maximal value,
- **Active pairs of loci:** it is the number of pair of loci for which disequilibrium can be calculated, that implies that the loci are polymorphic, e.g. loci that have at least two remaining alleles in the current sample and in frequency greater than the *min freq* threshold,
- **Allele number:** number of alleles present in the current sample for the selected loci, all alleles are considered and the minimal frequency for rare alleles does not apply here,
- **Allele proportion:** (C1) the current number of alleles on the number of alleles at step 0 (on all removable and forced units),
- **Allele freq proportion:** (C2) the sum for alleles present in the current sample of their frequencies at step 0 (on all removable and forced units).

A first line in red gives the values for the initial tree on the whole data set (all units and selected loci). The line in blue is for the step 0 on the active unit set (removable and forced units).



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

### **'Record sample' tab**

- **Sample size**

The user chooses the retained sample by its size. It is a value between the number of active units (without the excluded units) and the number of forced units or 3, the three last units, if there is no forced unit.

- **Identifier file**

A .DON file is selected to record the results of the procedure for each unit. It may be an existing .DON file where all units of the input file (.ARB file for Max length sub tree or .VAR file for Min SD subset) are necessarily present, the new fields will be added after the fields already existing. It may be also a new .DON file which is initialized with the units of the input file and if an identifier has been selected in **Units** sub menu, the first recorded field is this unit identifier.

- **Identifier**

Record fields in the selected .DON file.

- The first recorded field is the step at which the unit has been removed. All forced units take the last step as value. For excluded units the value is 0. This identifier is independent of the selected sample size and can be used to characterize any subset: for a size  $m$ , all units with a non-null value lower or equal to  $m$  are in the subset. The label of this field is by default 'Tree remove order' but it can be modified by the user.

- The second recorded field is proper to the subset corresponding to the selected 'sample size' value. It takes values:

- 'Excluded' if the unit was excluded (value 0 in the previous field)
- 'Removed' if the unit was removable and has been effectively removed,
- 'Kept' if the unit was removable but has been kept in this sub tree,
- 'Forced' if the unit was forced (kept in any case in the sub tree).

The label of this identifier is by default 'Tree sample\_m' for a sub tree of size  $m$ , but it can be modified by the user.

If several subsets of different size are recorded, 'Tree remove order' field will not be repeated.

- **Record Sub tree for Max length sub tree**

Ask for the name of a new .ARB file to record the sub tree corresponding to the selected 'sample size' value, and open automatically a window to display this tree.

# Export to 'PHASE' software

## Method

PHASE is a free software using Monte-Carlo Markov chain algorithm to estimate haplotypes for unphased diploid data, available from:

<http://www.stat.washington.edu/stephens/software.html>

The input files for PHASE are text files that can have three formats. Darwin exports files at the alternative format in which the genotypes are listed on a single line, locus by locus.



*Option -f1 must be specified in the command to run PHASE.*

The following example concerns a set of 4 units and 5 multi-allelic loci:

DARwin file in input:

5.0 - ALLELIC - 2										
4	5									
Units	M1-1	M1-2	M2-1	M2-2	M3-1	M3-2	M4-1	M4-2	M5-1	M5-2
1	8	8	13	13	10	10	8	8	7	7
2	8	8	13	13	10	10	8	8	7	7
3	8	8	13	13	99	99	8	8	7	7
4	8	8	13	13	10	10	8	8	7	7

PHASE file in output:

```
4
5
MMMMM
#1
8 13 10 8 7
8 13 10 8 7
#2
8 13 10 8 7
8 13 10 8 7
#3
8 13 -1 8 7
8 13 -1 8 7
#4
8 13 10 8 7
8 13 10 8 7
```

(see the documentation for PHASE 2.1)

## Procedure

### Input / output files

- . In input, a .VAR file (allelic data type) on diploids.
- . In output, a file in PHASE format with extension .INP.

## Missing data

Check the box to indicate that some data are missing in the allelic data file. The integer value used to code missing data has to be specified by the user (**Integer code for missing data**).

These missing will be coded with the convenient value for PHASE.

## Units

Selection of a subset of units for the PHASE file.

The currently unselected/selected units are listed in the left/right columns.



Buttons to move part or all units between columns

- **identifiers** to select a .DON file and an identifier in this .DON file if selection is easier using another unit identifier.
- **Statistics on current selection** open a window listing some synthetic parameters for each unit: numerical identifier, selected identifier, status, number of missing allele, of loci with at least one missing allele, of homozygote loci, of heterozygote loci.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Loci

Selection of the loci retained for the PHASE file.

The currently unselected/selected loci are listed in the left/right columns.



Buttons to move part or all loci between columns

- **Statistics on current selection** open a window listing synthetic parameters for each selected locus: locus identifier, number of missing units, number of alleles, min value and max value for allele code.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Record selected subset

If checked, this option creates a new .VAR file recording a data matrix with only units and loci selected by the user. The unit numerical identifiers are those of the initial .VAR file and any associated .DON file stays operational. By default, the proposed filename is *name\_sel.VAR* where *name* is the initial .VAR file name.

This option is very useful when a same tedious selection has to be used for several purposes.

# ***Import from 'PHASE' software***

## **Method**

Phase produces in output a text file compiling the results. DARwin extracts in this file the list of haplotypes inferred for each unit and record these data in a DARwin format.

In case of missing data, PHASE tries to infer the missing genotypes. In the output file, the uncertain genotypes are enclosed in [], depending on their certainty. The level of required certainty is controlled by the parameter `-q` in PHASE.

## **Procedure**

### **Input / output files**

- . In input, a PHASE output file,
- . In output, a .VAR file of allelic type on diploids with inferred haplotypes.

### **Missing data**

If values enclosed in brackets (estimated missing data) are found, two options are proposed:

- . **Accept missing data estimation**, as inferred by PHASE
  - . **Refuse missing data estimation**, these data will be coded as missing data in the DARwin file with a code specified by the user: **Integer code for missing data**.
-

# Tools

## *Random 0/1 data*

For methodological purposes, it may be useful to generate random binary variables. A random value is created in drawing a random number between 0 and 1, if this value is greater than .5, the variable is set to 1 and to 0 if not.

(see [Random dissimilarities](#) to generate quantitative random variables)

### Parameters

- Number of units
- Number of variables
- Random seed value
  - . Timer: references the system clock to initialize the seed, each run will give a different result.
  - . User-defined: a seed is specified to override the system clock, each run with the same specified seed will give same draw.

**Output file:** a .VAR file (type single) to record the generated data matrix. Successive variables are labelled as V-1, V-2...

**Output window:** display the resulting data if 'Display when done' is checked.

## *Random dissimilarities*

For methodological purposes, it may be useful to generate random dissimilarities with particular properties (see [Dissimilarity menu – Method](#) for property definitions) which are save in a .DIS file. For some dissimilarity types, a set of variables is firstly generated and is used in a second step to calculate the dissimilarities, it is proposed to save these data in a .VAR file.

### Parameters

- Number of units
- Dissimilarity type:
  - **Dissimilarity:** a basic dissimilarity without any other property
  - **Euclidean distance:** a set of  $q$  quantitative variables ( $q$  is the space dimension) is randomly generated. These variables will be saved if 'Record variables' is checked. The resulting dissimilarity is calculated as an Euclidean distance on these variables.
  - **$p$ -Minkowsky** distance: a set of  $q$  quantitative variables ( $q$  is the space dimension) is randomly generated. These variables will be saved if 'Record variables' is checked. The resulting dissimilarity is calculated as a Minkowsky distance of order  $p$  on these variables (= City-Block distance if  $p=1$ ).

- **Simple matching:** a set of  $q$  ( $q$  is the space dimension) 0/1 variables is randomly generated for each unit. These  $q$  variables will be saved if 'Record variables' is checked. The simple matching is calculated as the unmatching number between two units (01 or 10) standardized by the variable number.
- **Ultrametric distance:** a binary tree structure is randomly generated with random edge lengths such that the distance from each leaf to the root is constant.
- **Additive tree distance:** the sum of an ultrametric and a star distance is an additive tree distance; so an ultrametric is firstly generated and a random star distance is added. For a more accurate tree distance generation, use [Random tree](#) procedure and [Tree distance](#) to generate the corresponding distance matrix.
- **Robinson distance:** this very particular distance is linked to pseudo-hierarchies, called pyramids. Let  $v(i)$  be a valuation on each unit  $i$  and  $\leq$  an order on this valuation, then a Robinson distance verifies the condition:  $v(i) \leq v(j) \leq v(k) \Rightarrow d(i, k) \geq \max(d(i, j), d(j, k))$ .

- Random seed value

Timer: references the system clock to initialize the seed, each run will give a different result.

User-defined: a seed is specified to override the system clock, each run with the same specified seed will give same draw.

### Output files

- A .DIS file to record the generated dissimilarity matrix.
- If a set of  $q$  variables is firstly generated and if '**Record variables**' is checked, a .VAR file (type single) records these generated variables. It automatically receives the same name as the dissimilarity with extension .VAR. Successive variables are labelled as V-1, V-2... V- $q$ .

### Output window

Display the resulting dissimilarity matrix if 'Display when done' is checked.

## Random tree

For methodological purposes, it may be useful to generate random trees. Three types of tree are proposed:

- **Binary trees** where the degree of internal nodes is always equal to 3, with as many leaves as units
- **Complete trees** where some internal nodes may have a degree greater than 3, with as many leaves as units
- **General trees** where some internal nodes may have a degree greater than 3, some units may label internal nodes, so the number of leaves is lower than the number of units.

Tree generation starts from a tree on 3 units, the following units are successively randomly grafted on the tree.

- **Parameters**

- Tree type
  - . Binary
  - . Complete
  - . General
- Number of units
- Edge lengths
  - . All edges are set to 1
  - . Edge lengths are randomly generated in [0-1[
- Random seed value
  - . Timer: references the system clock to initialize the seed, each run will give a different result.
  - . User-defined: a seed is specified to override the system clock, each run with the same specified seed will give same draw.

- **Output file:** a .ARB file to record the generated tree.

- **Graphical window:** show the resulting tree if 'Display when done' is checked.

## ***Transpose data file***

This function creates from a .VAR file another .VAR file where rows and columns are exchanged. This function is mainly used when data come from spreadsheets like Microsoft Excel which is limited to 256 in column number. For large data set with more than 256 variables (and less than 256 units) a solution is to create a .VAR file with variables in rows and units in columns. **Transpose data file** function restores units in rows and variables in columns in a new .VAR file.

Example:

Input .VAR file with 6 rows and 5 columns and its associated .DON file

@DARwin 5.0 - VAR					
6	5				
N°	L1	L3	L7	L8	L10
2	475	90	250	30	140
3	10	10	495	110	170
6	585	115	50	0	150
11	656	97	97	10	52
12	168	22	528	69	102
20	615	125	95	0	115
@DARwin 5.0 - DON					
6	1				
N°	Name				
2	M1				
3	M2				
6	M3				
11	M4				
12	M5				
20	M6				



Output .VAR with 5 rows and 6 columns and its associated .DON file:

@DARwin 5.0 - VAR						
5	6					
Unit	M1	M2	M3	M4	M5	M6
1	475	10	585	656	168	615
2	90	10	115	97	22	125
3	250	495	50	97	528	95
4	30	110	0	10	69	0
5	140	170	150	52	102	115
File transposed: Example.var						

DARwin 5.0 - DON	
5	1
Unit	Name
1	L1
2	L3
3	L7
4	L8
5	L10

Units are identified by a sequential numerical identifier and correspondences with the initial unit names (column labels in the input file: L1, L3...) are recorded in an associated .DON file.

Variables can be identified by an external identifier selected in a .DON file. If no external identifier is defined, variable names are created as 'V-' + the row numerical identifier in the input file (V-2, V-3, V-6... in the previous example).

- **Input files:**

- a .VAR file (type single, allelic, sequences)
- optionally an associated .DON file where the variable labels will be read (if the .DON file contains several identifiers, the first one will be regarded as label for variables)

- **Output files:**

- A .VAR file (with the same type as the input file)
- The associated .DON file to record the unit labels.



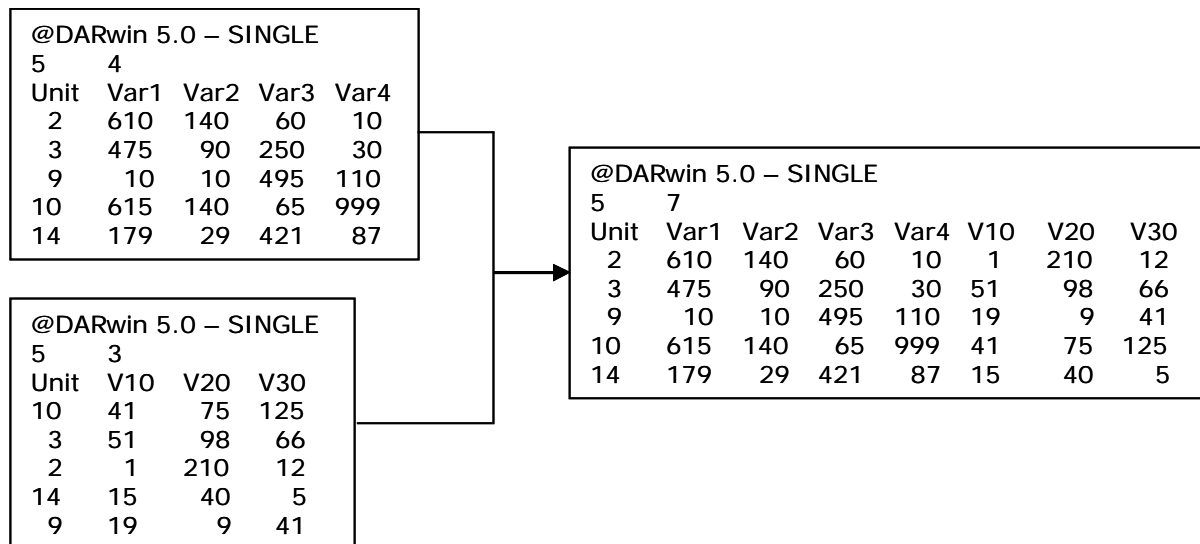
*For allelic data, the ploidy in the input file has to be set at the ploidy value required by the output file, even if this value has no meaning for the input file. The procedure verifies that the number of rows in the input file is a multiple of the ploidy.*

## **Merge data files**

This function creates a new .VAR file merging two .VAR files of same type (single, allelic, sequence). The procedure automatically proposes:

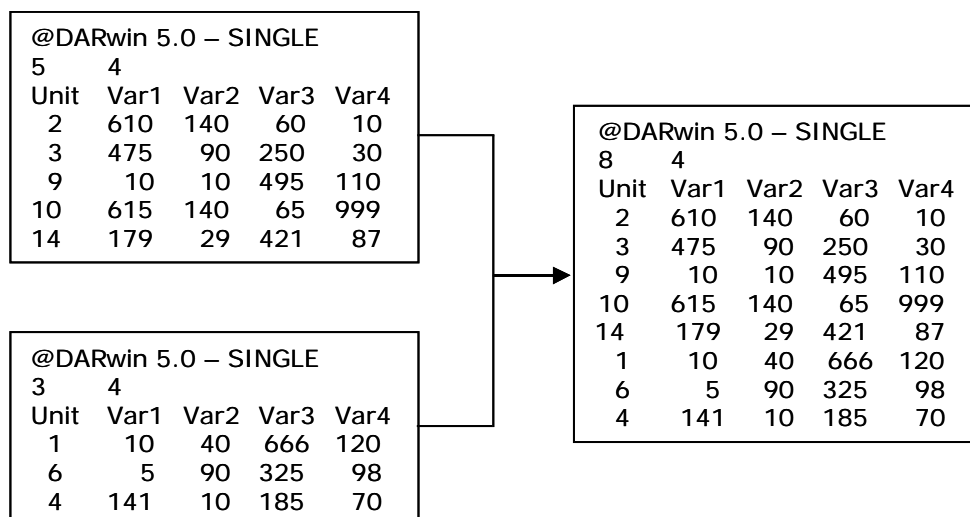
- **Horizontal** merging:

If the two files in input have the same number of units with identical numerical identifiers (not necessarily in the same order) but two different sets of variables, (no variable label in common) the output file merges the two sets of variables for each unit.



### - **Vertical** merging:

If the two files in input have two different sets of units (all the numerical identifiers are different) but the same number of variables, with the same labels and in the same order, the output file adds the second set of units at the end of the first one.



### **Input files:**

- Two .VAR files (type single, allelic, sequences)
- Optionally, for vertical merging, the associated .DON files.

### **Output files:**

- A .VAR file (with the same type as the input files)
- Optionally, for vertical merging, the associated .DON file that merges the list of identifiers for the first set of units and for the second set of units as read in the .DON files in input (these files in input may concern larger sets of units but only units in the two .VAR files will be retained).

## Single data correlations

Calculate correlation coefficients between pairs of variables read in a single data .VAR file.

**Input file:** a .VAR file (type single data)

Output file: [optional] to record the correlation matrix between variables as a .DIS file. This correlation after transformation ( $1-r^2$ ) can be used as a measure of dissimilarity between variables.

- A .DIS file. By default, the proposed filename is *name\_CorVar.DIS* where *name* is the initial .VAR file name.
- The associated .DON file is automatically created for recording names of variables.

### Missing data

Correlation between two variables can be evaluated only on units with valid values for the two variables, so any invalid value (missing data) has to be discarded from for these variables.

Three options are proposed to discard missing data:

- **Complete unit deletion**

Discard any unit with at least one missing data for one variable. The correlations will be calculated for all pairs of variables but only on the subset of units without any missing value.

- **Complete variable deletion**

Discard any variable with at least one missing data (this variable is discarded for all units). The correlations will be calculated on the whole set of units but only for variables without any missing value.

- **Pairwise deletion** (option by default)

If for a pair of variables, at least one of the two values is missing for a unit, this unit is discarded for computing correlation between these variables (and only for them). The correlations will be calculated for all pairs of variables but on a subset of units specific to the considered pair of variables.

A minimal proportion of valid units for a pair of variables can be chosen (50, 60, 70, 80 or 90%). For the first pair which does not reach this threshold, the procedure aborts and a window identifies the incriminate pair.

The two first options discard all missing data from the data set. They should be used when missing data are concentrated in some units or variables only. If missing data are distributed more or less at random, a great number of valid data may be also discard with complete deletion options. Pairwise deletion option avoids this loss of information in removing only missing data for the considered pair.

### Unit selection

This option allows computing correlations on a user defined subset of units.

The currently unselected/selected units are listed in the left/right columns.



Buttons to move part or all units between columns

- **Identifiers** to select a .DON file and an identifier in this .DON file if selection is easier on another unit identifier.

. **Statistics on current selection** open a window listing some synthetic parameters for each unit.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

### Variable selection

This option allows computing correlations on a user defined subset of variables. The currently unselected/selected variables are listed in the left/right columns.



Buttons to move part or all variables between columns

- **Statistics on current selection** open a window listing some synthetic parameters for each variable.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

### Text window

- Characteristics of the input file
- List of selected units
- Correlation matrix between variables



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## ***Re-label trees for common identifiers***

This function creates a common identifier file for several trees that share common units (not necessarily all the units) but where numerical identifiers are proper to each tree.

The common units are revealed by their identical label for a specified identifier in the .DON files associated to each tree.

The list of all labels found for this identifier is sorted and each occurrence receives an increasing numerical value.

Each initial tree is duplicated in a new tree file where these new numerical identifiers replace the initial numerical identifiers. A common identifier file is created which records the correspondence between the numerical code and the value for the common key identifier. Following fields (a field for each tree) code for presence (1) or absence (0) of a unit in each of these trees.

Example for two trees H1 and H2 by merging the two unit lists on the key identifier 'Name'. In output, the common H1H2.DON and the recoded trees H1\_R.ARB and H2\_R.ARB.

H1.DON

```
@DARwin 5.0 - DON
3      1
Unit   Name
2      CCC
4      AAA
5      FFF
```

H2.DON

```
@DARwin 5.0 - DON
5      1
Unit   Name
1      MMM
2      DDD
12     CCC
14     EEE
21     AAA
```

H1H2.DON

```
@DARwin 5.0 - DON
6      1
Unit   Common key H1-R  H2-R
1      AAA         1     1
2      CCC         1     1
3      DDD         0     1
4      EEE         0     1
5      FFF         1     0
6      MMM         0     1
```

Unit numerical identifiers in:

H1-R: 2, 1, 5

H2-R: 6, 3, 2, 4, 1

## Input / output files



to add a new tree file



to remove highlighted tree

For each added tree:

- The .ARB file
- The associated .DON file
- The key identifier in this .DON file (if the .DON file contains several identifiers)
- A name for the new .ARB file that records the recoded tree.

In output, a name for the common .DON file.

## Text window

- Displays the parameters on the input files
- Lists the numerical code associated to each occurrence of the key identifier.

# ***Pooling SSR alleles***

## **Method**

SSR markers are known to present a high mutation rate, classically described by a stepwise mutation model (gain or loss of one repeat unit at each mutation). So SSR markers may exhibit a high number of alleles when observed on a population covering a large genetic diversity (for example, up to 30 alleles for SSR markers on a collection of 660 sorghum accessions). This hypervariability is an advantage for some genetic applications but it becomes a severe limitation in other cases: phylogeny, association analyses...

A pragmatic solution should be to reduce the number of alleles by pooling alleles that bring in part the same information. For that, it is assumed that the observed allelic diversity results from (i) the genetic structure of the population due to demographic and breeding events, and (ii) secondary and recent mutational events following the SSR stepwise mutation model.

So, we retain a doubly mutational process:

- a primary process, generally by large insertions or deletions, has generated  $K$  classes of different allele size  $\mu_1, \dots, \mu_k, \dots, \mu_K$ .
- a secondary process, specific to SSR markers, has induced from an initial allele  $\mu_k$ , local variations around  $\mu_k$ . We assume that these variations can be approached, under a one step model, by a Gaussian distribution.

Only the primary process is considered as informative, secondary local variations being seen as a noise. The objective of this procedure is to infer the  $K$  classes and the alleles assigned to each one and to replace in the output file each allele value by its class.

It can be shown that under a stepwise model, the secondary distributions can be approximated by Gaussian distributions with the initial marker length as mean and the product mutation rate by time of divergence as variance.

Consequently, the observed distribution of alleles can be seen as a Gaussian mixture model which parameters (mean and variance of each Gaussian component and proportion of each component) have to be estimated. Mixture parameters cannot be directly achieved and an iterative Expectation-Minimization (EM) algorithm must be used.

For a given number of classes, the algorithm starts with a random initial assignment of alleles to classes and iteratively improves the solution to maximize the likelihood of the data. To avoid effect of initial random draw, several starting points are explored (see parameter 'random' in the procedure) and the better solution is retained.

The classical EM algorithm is based on the probability that an allele belongs to a class. Then an allele is not assigned to a single class but is assigned to every class with a probability. For the doubly mutational process that is retained here, this assignment in probability has a no real meaning and an allele is, or is not, in a class and it would be better founded to assign at each step an allele to only one class. This can be done using CEM algorithm, a classification version of the EM algorithm which adds a C-step (for Classification) assigning an allele to its class of maximal probability. Note that CEM does not maximise the same likelihood as EM and so does not converge necessarily to the same set of estimators. CEM is the option by default in the procedure.

A second issue of the double stochastic process concerns the definition of the assignment rule. The classical assignment rule means that a unit has more chances to belong to a frequent class rather than to a rare class and when means and variances of these classes are close, the class frequencies become the main factor of assignment decision. A consequence is to emphasize the frequent classes (and their variance) and to empty the uncommon classes. There is no reason in our case which a particular allele belongs to a class rather than to another. This leads to question the prior probability in the Bayes's rule. Instead of the frequencies of classes, the prior probability that an allele belongs to a class is chosen here as a uniform probability of one on the number of classes.

Another modification of the classical EM algorithm concerns the estimation of the variances which are specific to each component. In practice it appears that the algorithm may converge to unrealistic solutions with some classes centred on frequent alleles with very low variance and other classes encompassing a large range of less frequent alleles with very high variance. Assuming that the mutation rate cannot be very different between alleles, we prefer to fix a common variance for all components.

The algorithm returns the likelihood and the parameters of the better solution for a given number of classes but does not say anything on the best number of classes. The likelihood necessarily increases with the number of classes and is maximal for a number of classes equal to the number of alleles. So an other criterion must be used to select the optimal solution. We have retained here the Elbow Likelihood criterion ( $EL$ ) which measures the increase of the likelihood between steps  $K$  and  $K+1$  (in proportion of the likelihood  $L_1$ ):

$$EL_K = \frac{L_{K+1} - L_K}{L_1}$$

(in practice, we consider  $-EL_K$  to have a positive criterion).

When this increase becomes very low, it can be suspected that the addition of a new class does not improve the resolution. So the  $K$  value such that  $EL_K$  is minimum can be considered as the right solution. However this minimum is generally reached very slowly and the best solution is often for a lower  $K$  value. So the optimal solution is defined here as the first  $K$  such that:

$$-EL_K \leq -(1 + \frac{\alpha}{100})EL_{\min}$$

where  $\alpha$  is a user-defined parameter (if  $\alpha = 0$  it is the absolute minimum), by default, we have fixed  $\alpha = 1\%$  which gives often pertinent solutions.

## Marker selection

The first window proposes the list of SSR markers in a .VAR file in input.

### Input / output files

. A .VAR file (allelic data type) has to be selected in input. The reading procedure aborts if the type read in the file header does not correspond to the expected type.

. A name for the output .VAR file (allelic data type) has to be chosen. By default, the proposed filename is *name\_pool*.VAR where *name* is the initial .VAR file name.

### Unit selection

The currently unselected/selected units are listed in the left/right columns.



Buttons to move part or all units between unselected/selected columns.

- **Identifiers** to select a .DON file and an identifier in this .DON file if selection is easier using an another unit identifier.
- **Statistics on current selection:** open a window listing some synthetic parameters for each unit: numerical identifier, selected identifier, status, number of missing alleles, of loci with at least one missing allele, of homozygote loci, of heterozygote loci.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

### Locus selection

The currently unselected/selected loci are listed in the left/right columns.



Buttons to move part or all loci between unselected/selected columns.



- **Statistics on current selection:** open a window listing synthetic parameters for each selected locus: locus identifier, number of missing units, number of alleles, min value and max value for allele code.



**Print** the text window on the current printer



**Save** the text window in a .TXT or .RTF file.

## Missing data

Check the box to indicate that some data are missing in the allelic data file. The integer value used to code missing data has to be specified by the user (**code for missing data**).

## Record selected subset

If checked, this option creates a new .VAR file recording a data matrix with only units and variables selected by the user. The unit numerical identifiers are those of the initial .VAR file and any associated .DON file stays operational. By default, the proposed filename is *name\_sel*.VAR where *name* is the initial .VAR file name.

**OK** to update the list of markers after modifications of unit/locus selection and/or missing data options (at the opening, the list is displayed for no selection and no missing data).

## Marker list

For each marker, the number of presences (for example, twice the number of units for a diploid), the number of alleles, the number of classes after pooling (at the opening, is equal to the number of alleles), the number of alleles set to 'missing' (is null at the opening).

. **click** on a marker opens or updates a frame giving for each allele of the selected locus its value and the number of this allele in the dataset (or closes this frame if it is open for this locus).

## Allele list

. **double click** on an allele put this allele to 'missing' or conversely.

## Pooling button

Opens a specific window to define classes of non-missing alleles that have to be pooled.

## Record button

Record under the output filename the resulting data for the selected units and the selected markers. The initial allele values are replaced by the corresponding class. Initial missing data and alleles defined as missing keep the code for missing data (999 by default).

The correspondences between pooled classes and initial alleles are recorded in the comment field of the output file.

## Pooling alleles for a marker

At the opening, the algorithm is launched for the selected marker with the parameters by default. It returns for every number of classes from 1 to Kmaxi (see parameters) the assignment of alleles to classes that optimizes the likelihood.

### Parameters

The user can modify the parameters of the algorithm:

- . **EM** or **CEM** algorithm (by default CEM)
- . **Kmax**: maximum number of classes considered by the algorithm (1 to the number of alleles). By default, this value is the number of alleles if lower than 10 or 10 if not.
- . **random**: the number of random iterations for each class number (10 to 500 step 10, by default 100)
- . **alpha**: the parameter of the EL threshold to select the optimal number of classes (0 to 10 step 0.5, by default 1%)
- . **Run again**: to launch the algorithm with the new parameters

### LH graph

The algorithm results are summarized on the likelihood graph that displays for each number of classes the loglikelihood (red curve) and the EL criterion (blue curve). This graph does not depend on the selected number of classes; it is updated only if algorithm parameters are modified.

- . **-LHmax** gives the maximum on LH axis, this value can be changed to modify the scale on this axis.
- . **-ELmax** gives the maximum on EL axis, this value can be changed to modify the scale on this axis.

### Kc

This box gives the selected number of classes and the corresponding likelihood. By default it is the number of classes as defined by the EL criterion (this Kopt value is displayed beside the Kc box with its likelihood). The user can selected any other value for Kc between 1 and Kmaxi. The results are updated for this new Kc value in the allele table and on the Frequency graph.

### Allele table

This table lists the allele values, their frequency in the selected dataset and their class of assignment for the selected number of classes Kc.

#### . dble click

The limits of the classes can be manually modified by the user.

- A double click on the first allele of a class (if it is not the first one) moves this allele to the previous class.
- A double click on the last allele of a class (if it is not the last one) moves this allele to the following class.

If the class limits are modified, the representations of the classes on the frequency graph are updated. The likelihood is calculated for these new classes and displayed in the Kc box.

### Frequency graph

This graph displays the frequencies in vertical axis in function of the allele length in horizontal axis.

- . **max Freq** gives the maximum on the vertical axis, this value can be changed to modify the scale on this axis.
- . **Y step** fixes the step on EL axis, this value can be modified, for example, 2 for a dinucleotide SSR.

The graph shows also the current class definition: the alleles linked by a red line belong to the same class. The red triangle indicates the value that will be assigned to the class in the output file. It is the value of the most frequent allele of the class or, if several alleles are equally frequent, it is the value of the allele the closest to the mean of the class.

### Validate button

Record the current class definition for this marker, close this window and come back to the previous window to select another marker.

---

?

## *User's manual (PDF)*

Open a PDF version of this manual.

---

## *How to cite DARwin*

for a reference to methods:

Perrier, X., Flori, A., Bonnot, F. (2003). Data analysis methods. In: Hamon, P., Seguin, M., Perrier, X., Glaszmann, J. C. Ed., Genetic diversity of cultivated tropical plants. Enfield, Science Publishers. Montpellier. pp 43 - 76.

for the software itself:

Perrier X., Jacquemoud-Collet J.P. (2006). DARwin software  
<http://darwin.cirad.fr/>

---

## Bibliography

Bonnot, F., Guénoche, A., Perrier, X. (1996). Properties of an order distance associated with a tree distance. In: Diday, E., Lechevallier, Y., Optiz, O. Ed., *Ordinal and Symbolic Data Analysis*. Springer. Paris. 252-261.

Furnas, G. W. (1989). Metric family portraits. *Journal of Classification*. 6: 7-52.

Gascuel, O. (1997). Concerning the NJ algorithm and its unweighted version, UNJ. In : *Mathematical Hierarchies and Biology. DIMACS workshop, Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society. Vol 37: 149-170.

Gascuel, O. (2000). Data model and classification by trees: the minimum variance reduction (MVR) method. *Journal of Classification*. 17(1): 67-100.

Kubicka, E., Kubicki, G., Mc Morris, F. R. (1995). An algorithm to find agreement subtrees. *Journal of Classification*. 12: 91-99.

Makarenkov, V., Legendre, P. (2004). From a phylogenetic tree to a reticulated network. *Journal of Computational Biology*. 11 (1): 195-212.

Perrier, X. (1998). Analyse de la diversité génétique : mesures de dissimilarité et représentations arborées. Doctor thesis. Université Montpellier II. Montpellier. 192 p.

Saitou, N., Nei, M. (1987). The Neighbor-Joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*. 4(4): 406-425.

Sattath, S., Tversky, A. (1977). Additive similarity trees. *Psychometrika*. 42(3): 319-345.